

AD-A267 720



DTIC  
ELECTE  
AUG 10 1993  
S c D

2

93-18230



DISTRIBUTION STATEMENT A

Approved for public release  
Distribution Unlimited

93 8 5 049

**GENERAL PURPOSE AND SCOPE.** The Applied Computational Electromagnetics Society Journal hereinafter known as the ACES Journal is devoted to the exchange of information in computational electromagnetics, to the advancement of the state-of-the-art, and to the promotion of related technical activities. A primary objective of the information exchange is the elimination of the need to "re-invent the wheel" to solve a previously-solved computational problem in electrical engineering, physics, or related fields of study. The technical activities promoted by this publication include code validation, performance analysis, and input/output standardization; code or technique optimization and error minimization; innovations in solution technique or in data input/output; identification of new applications for electromagnetics modeling codes and techniques; integration of computational electromagnetics techniques with new computer architectures; and correlation of computational parameters with physical mechanisms.

**SUBMISSIONS CONTENT.** The ACES Journal welcomes original, previously unpublished papers, relating to applied computational electromagnetics.

Typical papers will represent the computational electromagnetics aspects of research in electrical engineering, physics, or related disciplines. However, papers which represent research in applied computational electromagnetics itself are equally acceptable.

For additional details, see "Information for Authors", elsewhere in this issue.

**SUBSCRIPTIONS.** All members of the Applied Computational Electromagnetics Society (ACES) who have paid their subscription fees are entitled to receive the ACES Journal with a minimum of two issues per calendar year. Current annual subscription fees are:

**AREAS:** U.S. and Canada: AIRMAIL: \$60 Individual, \$110 Organizational, SURFACE MAIL N/A; Mexico, Central & So. America: AIRMAIL: \$65 Individual, \$110 Organizational, SURFACE MAIL: \$63; Europe, Former USSR, Turkey, Scandinavia, AIRMAIL: \$73 Individual, \$110 Organizational, SURFACE MAIL: \$63; Asia, Africa, Middle East and Pacific Rim: AIRMAIL: \$80 Individual, \$110 Organizational, SURFACE MAIL: \$63. **FULL-TIME STUDENTS:** \$25.00. REMIT BY: (1) BANK DRAFTS (MUST BE DRAWN ON U.S. BANK), (2) INTERNATIONAL MONEY ORDER, (3) TRAVELER'S CHECKS IN U.S. DOLLARS, (4) ELECTRONIC TRANSFER, (CONTACT ACES EXECUTIVE OFFICER, RICHARD ADLER).

Back issues, when available, are \$15.00 each. Subscriptions to ACES, orders for back issues of the ACES Journal and changes of addresses should be sent to:

Dr. Richard Adler  
ACES Executive Officer  
ECE Department, Code ECAB  
Naval Postgraduate School  
833 Dyer Road, Room 437  
Monterey, CA 93943-5121 USA

Allow four week's advance notice for change of address. Claims for missing issues will not be honored because of insufficient notice or address change or loss in mail unless the secretary is notified within 60 days for USA and Canadian subscribers or 90 days for subscribers in other countries, from the last day of the month of publication. For information regarding reprints of individual papers or other materials, see "Information for Authors".

**LIABILITY.** Neither ACES or the ACES Journal editors are responsible for any consequence of misinformation or claims, express or implied, in any published material in an ACES Journal issue. This also applies to advertising, for which only camera-ready copies are accepted. Authors are responsible for information contained in their papers. If any material submitted for publication includes material which has already been published elsewhere, it is the author's responsibility to obtain written permission to reproduce such material.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By <i>Per A258 483</i>	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
<i>A-1</i>	

DTIC QUALITY INSPECTED 3

# APPLIED COMPUTATIONAL ELECTROMAGNETICS SOCIETY Journal

1993  
Vol. 8 No. 2

ISSN 1054-4887

The ACES Journal is abstracted in INSPEC, in Engineering Index, and in DTIC.

The second, third, fourth, and fifth illustrations on the front cover have been obtained from Lawrence Livermore National laboratory.

The first illustration on the front cover has been obtained from FLUX2D software, CEDRAT S.S. France, MAGSOFT Corporation, New York.

# THE APPLIED COMPUTATIONAL ELECTROMAGNETICS

## SOCIETY JOURNAL

### EDITORS

#### EDITOR-IN-CHIEF/ACES

David E. Stein  
Headquarters Air Force Sys. Command  
P.O. Box 169  
Linthicum Heights, MD 21090 U.S.A.

#### INTERIM EDITOR-IN-CHIEF/JOURNAL

Duncan C. Baker  
University of Pretoria/ECE Dept.  
0002 Pretoria, SOUTH AFRICA

#### MANAGING EDITOR

Richard W. Adler  
ECE Department Code ECAB  
Naval Postgraduate School  
Monterey, CA 93943-5121, U.S.A.

#### EDITOR-IN-CHIEF, EMERITUS

Robert M. Bevensee, SINGAPORE

Brian A. Austin  
University of Liverpool  
Liverpool, UK

Todd H. Hubing  
University of Missouri-Rolla  
Rolla, MO, U.S.A.

Kenzo Miya  
University of Tokyo  
Tokyo, JAPAN

Fulvio Bessi  
Ingegneria dei Sistemi S.p.A.  
Pisa, ITALY

Nathan Ida  
The University of Akron  
Akron, OH, U.S.A.

Osama A. Mohammed  
Florida International University  
Miami, FL, U.S.A.

Robert Bevensee  
National University of Singapore  
SINGAPORE

Magdy F. Iskander  
University of Utah  
Salt Lake City, UT, U.S.A.

Giorgio Mollinari  
University of Genova  
Genova, ITALY

John R. Bowler  
University of Surrey  
Surrey, UK

Kiyohiko Itoh  
Hokkaido University  
Sapporo, JAPAN

Frederic A. Mollinet  
Societe Mothesim  
Le Plessis-Robinson, FRANCE

Robert T. Brown  
Lockheed Aeronautical Sys. Co.  
Valencia, CA, U.S.A.

Randy J. Jost  
SRI International  
Arlington, VA, U.S.A.

Gerrit Mur  
Technische Universiteit Delft  
Delft, NETHERLANDS

Chalmers M. Butler  
Clemson University  
Clemson, SC, U.S.A.

Linda P. B. Katchi  
University of Michigan  
Ann Arbor, MI, U.S.A.

Takayoshi Nakata  
Okayama University  
Okayama, JAPAN

Edgar Coffey  
Advanced Electromagnetics  
Albuquerque, NM, U.S.A.

Adalbert Konrad  
University of Toronto  
Toronto, Ontario, CANADA

Andrew F. Peterson  
Georgia Institute of Tech.  
Atlanta, GA, U.S.A.

Tony Fleming  
Telecom Australia  
Clayton, Victoria, AUSTRALIA

Peter Krylstedt  
National Defence Research Est.  
Sundbybygd, SWEDEN

Harold A. Sabbagh  
Sabbagh Associates, Inc.  
Bloomington, IN, U.S.A.

Pat Foster  
Microwave & Antenna Systems  
Gt. Malvern, Worc. UK WR14 4AP

Stanley Kubina  
Concordia University  
Montreal, Quebec, CANADA

Chris Smith  
Kaman Sciences Corp.  
Colorado Springs, CO, USA

Gregory R. Haack  
DSTO  
Salisbury, AUSTRALIA

Karl J. Langenberg  
Universitat Kassel  
Kassel, GERMANY

C. W. "Bill" Trowbridge  
Vector Fields Limited  
Oxford, UK

Christian Hafner  
Swiss Federal Institute of Technology  
Zurich, SWITZERLAND

David A. Lowther  
McGill University  
Montreal, Quebec, CANADA

Jean-Claude Verite  
Electricite de France  
Clamart, Cedex, FRANCE

Roger Harrington  
Syracuse University  
Syracuse, NY, U.S.A.

Ronald Marhefka  
Ohio State University  
Columbus, OH, U.S.A.

W. Perry Wheless  
University of Alabama  
Tuscaloosa, AL, U.S.A.

Donald F. Herrick  
ERIM  
Ann Arbor, MI, U.S.A.

Gerald Meunier  
INPG/ENSIEG  
St. Martin-d'heres Cedex, FRANCE

Keith W. Whites  
University of Kentucky  
Lexington, KY, U.S.A.

Kueichien C. Hill  
Wright Laboratory  
Wright-Patterson, AFB, OH, U.S.A.

Edmund K. Miller  
Los Alamos National Lab.  
Los Alamos, NM, U.S.A.

John W. Williams  
Science Applications Int.  
Albuquerque, NM, U.S.A.

Manfred Wurm  
FB Technik  
Kiel, GERMANY



# THE APPLIED COMPUTATIONAL ELECTROMAGNETICS

## SOCIETY JOURNAL

Vol. 8 No. 2

1993

### TABLE OF CONTENTS

"Computational Simulation of Electric Fields Surrounding Power Transmission and Distribution Lines" by J. P. Donohoe, M. Y. Jiang, J.F. Thompson, and D.B. Miller .....	4
"Comparison of Electromagnetic Response in Time and Frequency Domains" by S. Kashyap and A. Louie .....	17
"Validation of the Numerical Electromagnetics Code (NEC) for Antenna Wire Elements in Proximity to Earth" by M.M. Weiner .....	44
"Propagation of VLF Radiation in the Earth-Ionosphere Waveguide Excited by An Airborne Dual Trailing Wire Antenna" by D.H. Werner, J.K. Breakall, and R.J. Lunnen .....	72
"An Investigation of Wire Grid and Surface Patch Modeling Using the Numerical Electromagnetics Code (NEC)" by J.K. Breakall, R.W. Adler and P.D. Elliniadis .....	93
"An Integrated Environment for the Numerical Modeling of Communication Antennas Based on Relational Databases" by V. Stover and J.C. Logan .....	114
"An Application of the Hybrid Moment Method/Green's Function Technique to the Optimization of Resistive Strips" by R.C. Baucke .....	128
"Parallel Matrix Solvers for Moment Method Codes for MIMD Computers" by D.B. Davidson .....	144
"'A Priori' Knowledge, Non-Orthogonal Basis Functions, and Ill-Conditioned Matrices in Numerical Methods" by Ch. Hafner .....	176
"On the Convergence of the Method of Moments, the Boundary-Residual Method, and the Point-Matching Method with a Rigorously Convergent Formulation of the Point-Matching Method" by K.J. Bunch and R.W. Grow .....	188
"Solution of Team Benchmark Problem #10 (Steel Plates Around a Coil)" by O. Biro .....	203
"Solution of Team Benchmark Problem #13 (3-D Nonlinear Magnetostatic Model)" by O. Biro, Ch. Magele, and G. Vrisk .....	216
"Solution of Team Benchmark Problem #13 (3-D Nonlinear Magnetostatic Model)" by L. Kettunen and L. R. Turner .....	226
"Solution of Team Benchmark Problem #9 Handling Velocity Effects with Variable Conductivity" by K. Davey .....	232
Institutional Membership .....	244

# COMPUTATIONAL SIMULATION OF ELECTRIC FIELDS SURROUNDING POWER TRANSMISSION AND DISTRIBUTION LINES

J. Patrick Donohoe<sup>\*</sup>  
Min-Yee Jiang<sup>\*\*</sup>  
Joe F. Thompson<sup>\*\*</sup>  
David B. Miller<sup>\*</sup>

<sup>\*</sup>Department of Electrical and Computer Engineering  
Mississippi State University  
P.O. Drawer EE  
Mississippi State, MS 39762

<sup>\*\*</sup>Engineering Research Center for Computational Field Simulation  
Mississippi State University/National Science Foundation  
P.O. Box 6176  
Mississippi State, MS 39762

## ABSTRACT

*A numerical technique is presented for computing the potential distributions surrounding power transmission and distribution lines of complex geometry. The technique employs a finite difference solution using boundary-fitted coordinates. A newly developed finite difference solver code is coupled with the existing EAGLE grid generation code to yield a system capable of solving for the electric potential and field distributions surrounding complex configurations. A code validation example is presented which consists of a sphere-to-ground electrostatic solution. Sample results are also presented for a distribution line model.*

## INTRODUCTION

The voltage level at which a power transmission or distribution line fails (the critical flashover voltage) is dependent on the physical configuration of the line under test. The cross-sectional geometries of lines in use today vary widely given various supporting structures, insulators and conductors. The critical flashover voltage of a newly-designed configuration is commonly determined through construction and experimental testing. In some cases, attempts at reproducing experimental results fail because the arc traverses different paths to ground from test to test. In the design of transmission and distribution lines, prediction of the failure point is often accomplished by comparing the new configuration to a geometrically similar configuration which has previously been tested. A more effective method of predicting the failure point of a given configuration is to determine the potential and field distribution through computational means. This technique would greatly enhance the design of high voltage transmission and distribution lines by providing the designer a tool to investigate changes in the insulation properties of a given line due to minor design modifications without expensive experimental tests. An accurate plot of the potential and field distributions surrounding the line would yield insight into the maximum allowable voltage levels and the phenomenon of multiple paths to ground from failure to failure.

Computation of the electric potential distribution throughout some arbitrarily shaped two-dimensional or three-dimensional region involves the numerical solution of the governing partial differential equation. Since high voltage transmission and distribution lines carry either direct current or low frequency (60 Hz) alternating current signals, the potential distribution for breakdown calculations may be determined assuming no time variation. Under static conditions, the potential distribution is governed by either Poisson's or Laplace's equation, depending on the distribution of free electric charge in the region of interest [1]. The periodic placement of supporting structures along the length of any transmission or distribution line makes the problem of modelling such a configuration inherently three-dimensional.

The representation of the surface boundary conditions is a critical factor in the accuracy of the final solution to a given partial differential equation. The accurate numerical representation of surface boundary conditions for a transmission or distribution line with a complex supporting structure is by no means trivial. A particularly effective technique of accurately describing boundary conditions on an arbitrarily shaped body is through boundary fitted coordinates (numerical grid generation) [2], [3], [4]. A curvilinear coordinate system is defined in the region of interest such that all boundaries in the region are coincident with coordinate lines. The coordinate system describing the physical region is then transformed into a fixed rectangular computational field defined by a square mesh. The resulting system of finite difference equations in the transformed or computational space consists of simply described boundary conditions. The equations to be solved in the computational space are more complex than those in the original physical space, but the accuracy of the solution is enhanced through the precise representation of the surface boundary conditions in the computational space. The finite difference solutions to the equations in the computational space are obtained using only grid points so that no interpolation between grid points is required. The grid point placement is dictated by the field variation in the region of interest. Grid points are concentrated in regions where the field variation is rapid while widely-spaced grid points are used in regions where the field is near-constant.

The electric potential distributions computed in this research were obtained using a newly developed solver code coupled with the existing EAGLE grid generation code [5] to yield a system capable of solving for potential distributions surrounding transmission and distribution line configurations defined by complex geometries. The grid system in the region surrounding the transmission line model of interest is constructed using the EAGLE code. The finite difference method is utilized to solve the governing partial differential equation over the domain of interest subject to the appropriate boundary conditions.

## FORMULATION

Given some static distribution of electric charge, the resulting electric scalar potential ( $V$ ) may be determined as a function of position by solving the appropriate boundary value problem. The differential equation which describes the potential distribution in a region of (free) electric charge is Poisson's equation given by

$$\nabla^2 V = -\frac{\rho}{\epsilon} \quad (1)$$

where  $\nabla^2$  is the Laplacian operator,  $\rho$  is the electric charge density ( $C/m^3$ ) and  $\epsilon$  is the permittivity ( $F/m$ ) of the medium. In a region where no free charge is present, Poisson's equation reduces to Laplace's equation:

$$\nabla^2 V = 0 \quad (2)$$

The solutions to Poisson's and Laplace's equations are obtained by enforcing the appropriate boundary conditions to the general solution of the respective differential equation. These boundary conditions may be expressed in terms of the scalar electric potential, the vector electric field  $E$  ( $V/m$ ) or the vector electric flux density  $D$  ( $C/m^2$ ). The static electric field is defined in terms of the electric scalar potential by

$$\mathbf{E} = -\nabla V \quad (3)$$

where  $\nabla$  is the gradient operator. Thus, boundary conditions described in terms of the vector electric field may be related to the electric scalar potential using Equation (3). The general equations which describe the behavior of the electric field and flux across a surface discontinuity are well known [1,6,7] and are given by

$$\mathbf{n} \times [\mathbf{E}_2 - \mathbf{E}_1] = 0 \quad (4)$$

and

$$\mathbf{n} \cdot [\mathbf{D}_2 - \mathbf{D}_1] = \rho_s \quad (5)$$

where  $(\mathbf{E}_1, \mathbf{D}_1)$  are vector quantities in region 1,  $(\mathbf{E}_2, \mathbf{D}_2)$  are vector quantities in region 2, and  $\mathbf{n}$  is a unit normal to the surface which points into region 1. The vector electric flux density is related to the vector electric field by

$$\mathbf{D} = \epsilon \mathbf{E} \quad (6)$$

The boundary conditions in Equations (4) and (5) can be related to the scalar potential using Equations (3) and (6) which yields

$$\mathbf{n} \times [(\nabla V)_1 - (\nabla V)_2] = 0 \quad (7)$$

and

$$\mathbf{n} \cdot [\epsilon_1 (\nabla V)_1 - \epsilon_2 (\nabla V)_2] = \rho_s \quad (8)$$

Equations (7) and (8) represent the general boundary conditions for the scalar potential across a surface discontinuity.

In cases where an isolated conductor is located in an applied electric field, the resulting conductor potential is unknown ("floating" conductor). When a conductor is placed in a static electric field, a charge distribution is induced on the conductor surface which produces a zero-valued electric field everywhere inside the conductor yielding a constant-valued potential throughout the conductor. The total surface charge on the conductor remains unchanged given any applied electric field distribution. From Gauss' Law, the integral of the normal component of the electric flux density over the outer surface of the charged conductor ( $S$ ) yields the total charge on the conductor such that

$$\int_S \mathbf{D} \cdot d\mathbf{s} = Q \quad (9)$$

where the direction of  $d\mathbf{s}$  is an outward pointing normal ( $d\mathbf{s} = \mathbf{n} ds$ ),  $ds$  defines the differential surface element on the conductor surface and  $Q$  is the initial value of total charge on the conductor. The integral defined in Equation (9) may be expressed in terms of potential by relating the electric flux density to the potential which yields

$$\int_S \epsilon [\mathbf{n} \cdot (\nabla V)] ds = -Q \quad (10)$$

where the dielectric surrounding the conductor is assumed to be isotropic.

Assuming the transmission or distribution line model is composed of perfect conductors and lossless dielectrics, the regions of interest with regard to potential and field computations (external to the conductors and throughout the dielectrics) are charge-free. Thus, a solution to Laplace's equation is required. Given a curvilinear coordinate system defined by  $(\xi^1, \xi^2, \xi^3)$ , the Laplacian operator in non-conservative form [2] is given by

$$\nabla^2 V = \sum_{i=1}^3 \sum_{j=1}^3 g^{ij} V_{\xi^i \xi^j} + \sum_{k=1}^3 (\nabla^2 \xi^k) V_{\xi^k} \quad (11)$$

where the subscripts on  $V$  denote partial derivatives and  $g^{ij}$  is the contravariant metric tensor. The elements of the contravariant metric tensor are defined as dot products of the contravariant unit vectors (normals to the curvilinear coordinate surfaces denoted by  $\hat{a}^i$ ) which yields

$$g^{ij} = \hat{a}^i \cdot \hat{a}^j \quad . \quad (12)$$

The Laplacian term in Equation (11) may be written as

$$\nabla^2 \xi^k = g^{kk} P_k \quad (13)$$

where  $P_k$  is a control function evaluated in the course of the grid generation and is then available to the Laplace solver as coefficients with fixed values at each grid point. Laplace's equation can now be written as

$$\sum_{i=1}^3 \sum_{j=1}^3 g^{ij} V_{\xi^i \xi^j} + \sum_{k=1}^3 g^{kk} P_k V_{\xi^k} = 0 \quad . \quad (14)$$

The first and second order derivatives found in Equation (14) are represented by central difference approximations and the overall equation is solved using the successive over-relaxation (SOR) iterative technique.

Four basic boundary conditions on the scalar potential are required in the formulation of the numerical model: the Dirichlet boundary condition, the Neumann boundary condition, the material interface boundary condition and the floating conductor boundary condition. The Dirichlet boundary condition is characterized by a scalar potential which is constant on a particular boundary. The surface of all conductors are defined as Dirichlet boundaries since they represent equipotential volumes. The application of the Dirichlet boundary condition is trivial using boundary-fitted coordinates. Scalar potential values are fixed at grid points on the specified surface ( $\xi^i = \text{constant}$ ) and these values are preserved throughout the iterative process.

The Neumann boundary condition is defined by a zero-valued normal derivative of the potential on a given boundary. The Neumann boundary condition is applied at the outer boundaries of the volume enclosing the transmission line model. The normal derivative to the coordinate surface on which  $\xi^i$  is constant is given by

$$V_n^i = \frac{\partial V}{\partial n} = \frac{1}{\sqrt{g^{ii}}} \sum_{j=1}^3 g^{ij} V_{\xi^j} \quad (15)$$

which yields

$$\sum_{j=1}^3 g^{ij} V_{\xi^j} \quad (\xi^i = \text{constant}) \quad (16)$$

as the Neumann boundary condition on the given surface.

A floating conductor is an equipotential volume but the conductor potential is an unknown value. Thus, the initial charge condition defined in Equation (10) must be coupled with the Dirichlet boundary condition for a floating conductor. Such a boundary condition can be enforced by integrating the normal component of electric flux over the conductor surface or over some surface enclosing the conductor. Both methods of integration produce similar results but the structure of the corresponding grids are totally different. Integrating over a surface enclosing the conductor allows for a multi-block grid with a smaller number of blocks than integrating over the conductor surface. The advantage of using a small block system is to obtain better control of the grid distribution in the regions of interest and to reduce the I/O overhead needed to transfer iterative data from block to block.

The material interface boundary conditions are applied at dielectric-dielectric interfaces and conductor-dielectric interfaces. The tangential electric field boundary condition of Equation (7) is implicit in the formulation since the scalar potential is assumed to be continuous across the boundary. The normal electric flux boundary condition of Equation (8) on the coordinate surface where  $\xi^i$  is constant may be stated as

$$\frac{\epsilon_1}{\sqrt{g^{ii}}} \sum_{j=1}^3 g^{ij} V_{\xi^j} - \frac{\epsilon_2}{\sqrt{k^{ii}}} \sum_{j=1}^3 k^{ij} V_{\xi^j} = \rho, \quad (17)$$

where  $g^{ij}$  and  $k^{ij}$  are the contravariant metric tensors evaluated on the grid at the interface in region 1 and region 2, respectively.

### COMPOSITE-BLOCK GRID STRUCTURE

The curvilinear coordinate system mentioned in the previous section is constructed using the EAGLE grid generation code. The EAGLE code is a composite (multi-block) algebraic or elliptic grid generation system designed to discretize the domain in and around any arbitrary shaped three-dimensional region. The concept of the composite-block structure is described in detail in [5].

Fundamental to the curvilinear coordinate system is the coincidence of some coordinate surface with each boundary of the physical region. The physical region of interest is divided into contiguous subregions (interfacing hexahedrons), and each subregion can be transformed to a rectangular block in the computational space, with a grid generated within each subregion. Each subregion has its own curvilinear coordinate system irrespective of that in the adjacent subregions. Each subregion, defined by six generally curved sides, is transformed to a rectangular computational region on which the curvilinear coordinates are the independent variables. In principle, it is possible to establish a correspondence between any physical region and a rectangular block in the computational space, but the resulting grid may be too skewed for a complicated geometry. In such a case, the given physical region must be subdivided into smaller blocks until the resulting grid satisfies the user-defined grid criterion with regard to skewing.

The general curved surfaces bounding the sub-regions in the physical space form internal interfaces across which information must be transferred. In the computational space, the information must be transferred from the side of a given rectangular block to the corresponding side of the adjacent block. These two sides of the adjacent blocks correspond to the same physical surface. The interface is treated as a branch cut on which the function value is solved just as it is in the interior of the blocks. The interfaces of the blocks are not fixed, but are determined by the solver. The most straightforward technique to employ is to provide an extra layer of points surrounding each block. These surrounding points represent points across the given interface just inside the adjacent block. This relationship is maintained throughout the iterative procedure. The governing partial differential equation is solved by point SOR iteration using a field of locally optimum acceleration parameters. These optimum parameters make the solution robust and capable of convergence with strong control functions.

### CODE VALIDATION EXAMPLE

An analytically solvable electrostatic problem is attempted in order to validate the code. The problem considered here is that of a conducting sphere over ground. The sphere over ground problem is solved numerically and compared with the equivalent problem of two isolated conducting spheres. For the two sphere problem, the electric field along the line connecting the sphere centers may be expressed as an infinite series using images as given in [8]. This problem has similar characteristics to the power distribution line problem in that one is interested primarily in the electric field and potential in the region between the conductors while these quantities are less critical away from this region. Also, the outer

boundary of the sphere-to-ground problem has the same general characteristics of the transmission/distribution line problem with a ground plane on the floor of the region and the remainder of the outer boundary on which the potential is unknown.

The geometry of the sphere-to-ground problem is shown in Figure 1 where  $D$  is the sphere diameter,  $S$  is the spacing between the sphere surface and the ground plane, and  $B$  is the dimension of the cubical outer boundary. The particular geometry chosen is  $D=100\text{cm}$ ,  $S=50\text{cm}$  and  $B=20D$ . Note that the  $x$  coordinate origin is located at the sphere center and extends downward to the ground plane. Thus, the domain of interest for field comparison purposes is  $50\text{cm} \leq x \leq 100\text{cm}$ . A three block, h-type grid is generated in the given volume of interest with a total of 88,263 grid points. Comparisons of the computed electric field with analytic results are shown in Figures 2 and 3 using two distinct outer boundary conditions. A constant potential of  $V=0$  is assumed on the ground plane and on all outer boundaries for the first boundary condition (Figure 2). For the second boundary condition (Figure 3), the ground plane potential is again assumed to be  $V=0$  but the Neumann boundary condition is enforced on the other outer boundaries. In both cases, excellent agreement is found between the analytic and computed results.

The uniform  $V=0$  outer boundary condition described above is viable for problems involving conductors of limited extent in all three dimensions such as the sphere-to-ground problem. However, for problems involving conductors which span the entire spatial domain in one or more dimension, such as a transmission or distribution line model, the constant outer boundary condition is inadequate and the Neumann boundary condition may be applied. The Neumann boundary condition, which forces the computed equipotential contours to lie normal to the outer boundary, yields appropriate behavior in the vicinity of the ground plane where the equipotential contours "follow" the ground plane. Negligible errors in the electric field are experienced in the regions of interest by choosing the outer boundaries sufficiently far away as shown in the sphere-to-ground example.

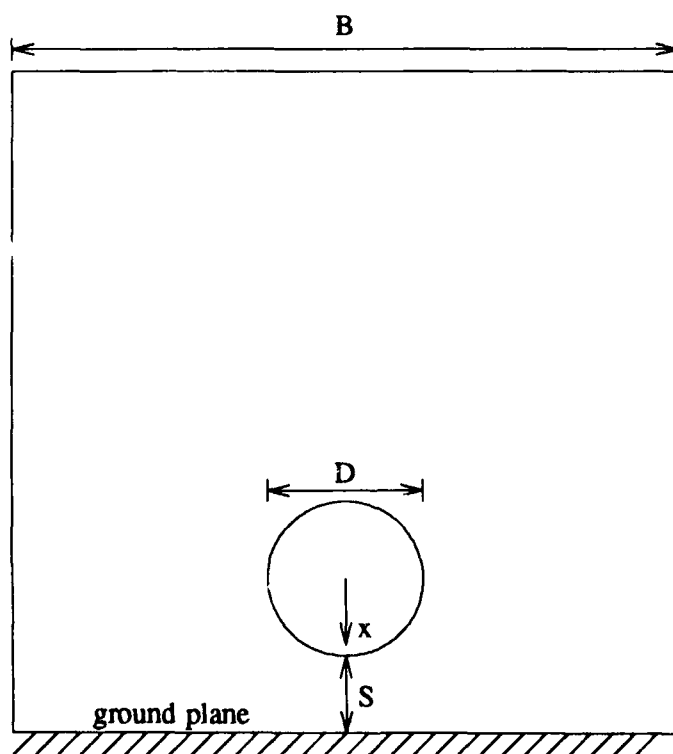


Figure 1. Sphere-to-ground geometry.

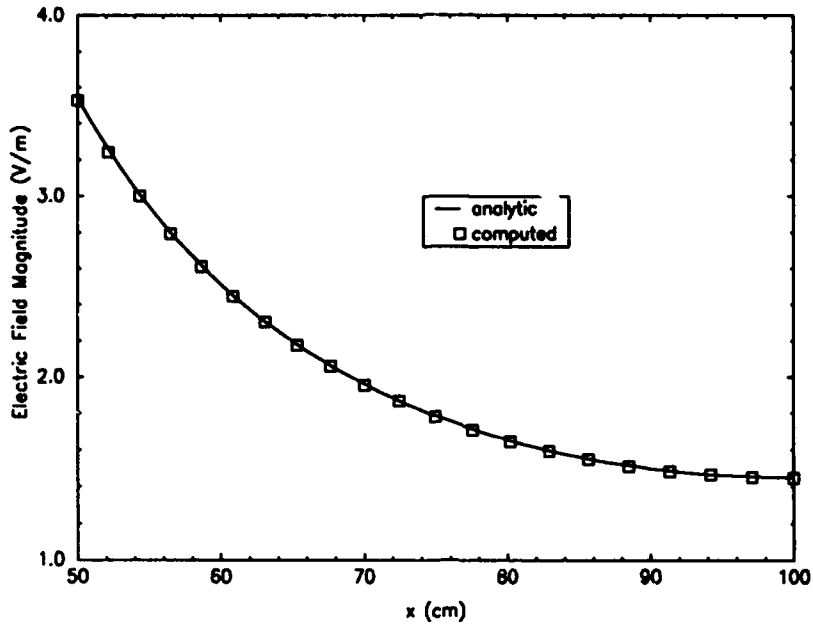


Figure 2. Comparison of the analytical sphere-to-ground electric field with computed results given  $V=0$  on all outer boundaries.

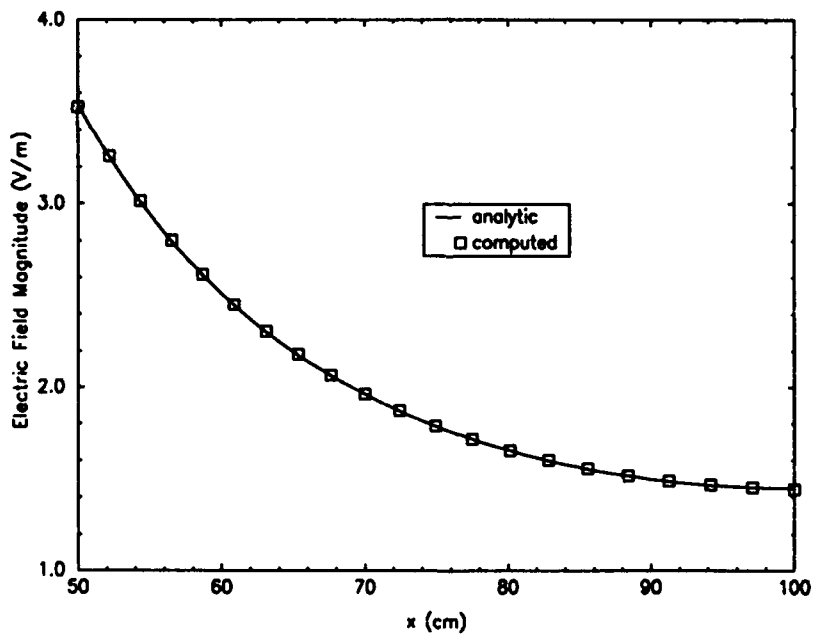


Figure 3. Comparison of the analytical sphere-to-ground electric field with computed results using the Neumann boundary condition on all outer boundaries excluding the ground plane.

### **POWER DISTRIBUTION LINE MODEL**

The critical flashover voltage of a typical distribution line configuration was studied experimentally by Jacob, et. al. in [9]. The distribution line model shown in Figure 4 is a simplified version of the experimental configuration analyzed in the aforementioned study. Insulators, crossarm braces and



mounting hardware components have been omitted in the distribution line model to simplify the geometry of the resulting grid. A detailed description of the distribution line numerical model is given in Table 1. Conductors A, B and C represent the three high-voltage conductors (phases) while conductor N is the neutral wire and conductor G is the vertical ground wire.

Several configurations of charged, floating and grounded phases were studied experimentally in [9]. A single phase was charged in order to measure the critical flashover voltage of the phase-to-ground and phase-to-phase failures. Two of the models considered in [9] are analyzed here and designated as model #1 and model #2. For model #1, the B phase is charged with the A and C phases floating. For model #2, the B phase is charged while the A phase is grounded and the C phase is floating. The ground plane, conductor N and conductor G are held at 0 volts for both models. The potential of the charged conductor is assumed to be 1 volt for both models.

The volume enclosing the three-dimensional distribution line model is defined by  $(-400'' \leq x \leq 200'')$ ,  $(0'' \leq y \leq 1200'')$  and  $(-90'' \leq z \leq 90'')$  with the axis of the pole located along the y-axis. A four-block grid system is constructed with the grid points distributed throughout the volume as illustrated in Figure 5. Note that the grid points are concentrated in the region of interest surrounding the conductors in the vicinity of the pole. A total of 96,525 grid points were used to determine the three-dimensional potential distributions: 75 in the x-direction, 39 in the y-direction and 33 in the z-direction. The computed potential distributions are plotted over surfaces defined by  $k=\text{constant}$  where  $k$  is the grid point index in the z-direction ( $k=0$  defines the plane where  $z=-90''$ ,  $k=33$  defines the plane where  $z=90''$ ). Due to the grid structure, the surfaces defined by  $k=\text{constant}$  are not planar in the vicinity of the pole as shown in Figure 6.

## RESULTS

The potential distributions are computed for both model #1 and model #2 given the charged conductor (phase B) is charged to 1 volt. The resulting potential distributions are plotted over constant  $k$  surfaces in the vicinity of the charged conductor. The potential distributions surrounding model #1 for  $k=1$ ,  $k=17$  and  $k=19$  are shown in Figures 7, 8 and 9, respectively. The corresponding potential distributions surrounding model #2 for  $k=1$ ,  $k=17$  and  $k=19$  are shown in Figures 10, 11 and 12, respectively. The potential difference between the equipotential contours is 0.02 volts.

The initial "guess" for the potential values has a significant effect on the number of iterations required for a prescribed accuracy. The first results were obtained by assuming a zero-valued potential at all grid points as the starting values. A significant reduction in the run time was obtained for both three-dimensional models by utilizing the corresponding two-dimensional results as the initial values on the surfaces normal to the axes of the wires. The two-dimensional results are those associated with the same distribution line minus the pole, crossarm and vertical ground wire. As expected, the potential distribution of the three-dimensional model approaches that of the two-dimensional model as one moves away from the supporting structure.

Several physical effects associated with discrete components of the numerical model have been noted for the configurations which were analyzed. The effect on the potential distribution of a floating conductor of small cross-sectional dimension is found to be minimal. Conversely, a floating conductor can alter the potential distribution considerably if the conductor cross-section is of significant physical dimension. The floating conductor represents an equipotential volume (surface) so that the resulting equipotential contours in the surrounding medium must wrap around the conductor. The equipotential contours on the end faces of the volume enclosing the three-dimensional model over perfect ground are predominantly horizontal below the charged conductors since the contours must "follow" the ground plane (equipotential surface). The effect of a vertical ground wire located on the supporting structure is to pull the equipotential contours upward as one moves from the end face toward the supporting structure such that the equipotential contours wrap around the ground wire. A large electric field is generated in the vicinity of the ground wire as the equipotential contours crowd together. The effect of the wood pole is

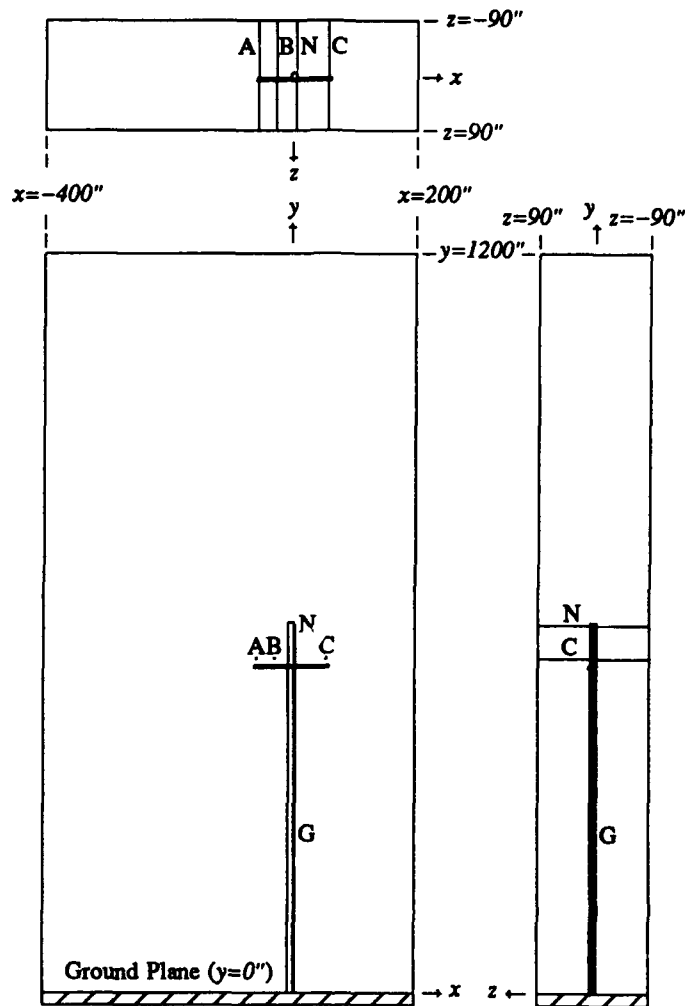


Figure 4. Three-dimensional distribution line model.  
(See Table 1 for detailed description.)

**Enclosing Volume** -  $[-400 \leq x \leq 200, 0 \leq y \leq 1200, -90 \leq z \leq 90]$

**Ground Plane** - Perfectly conducting ground plane (x-z plane)

**Pole** - Southern pine ( $\epsilon=3.5\epsilon_0$ ) 50 ft. pole, 10" diameter, axis of the pole lies along the y axis

**Crossarm** - Southern pine ( $\epsilon=3.5\epsilon_0$ ) 10 ft. crossarm, 120" x 4.5" x 5", centerline of crossarm is 44 ft. above the ground plane

**Conductors** - Each conductor is modelled as a filament,  
 A Phase - wire axis located at  $x=-56, y=541.25$   
 B Phase - wire axis located at  $x=-27, y=541.25$   
 C Phase - wire axis located at  $x=56, y=541.25$   
 Neutral (N) - wire axis located at  $x=5, y=595$   
 Ground wire (G) - wire axis located at  $x=5, z=0$

Table 1. Description of the distribution line model.

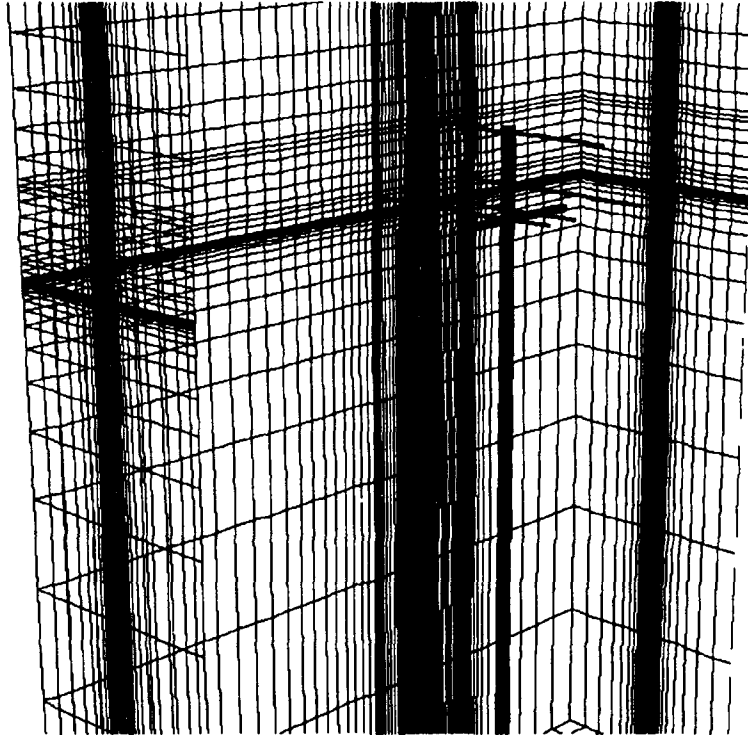


Figure 5. Three-Dimensional Distribution Line Model Grid.

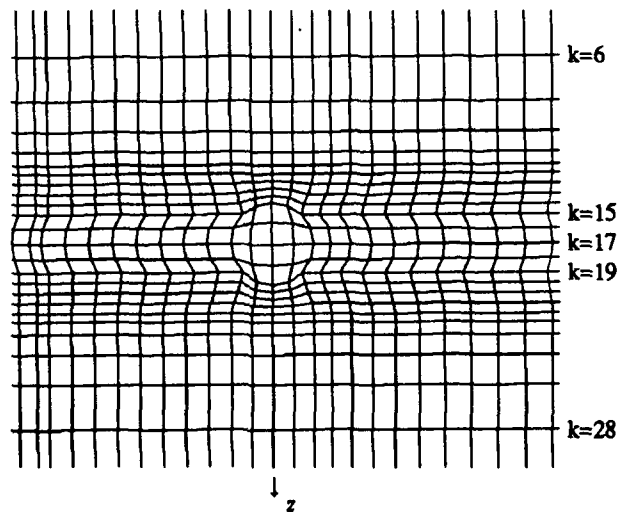


Figure 6. Cross-Sectional Contours in the Vicinity of the Pole.

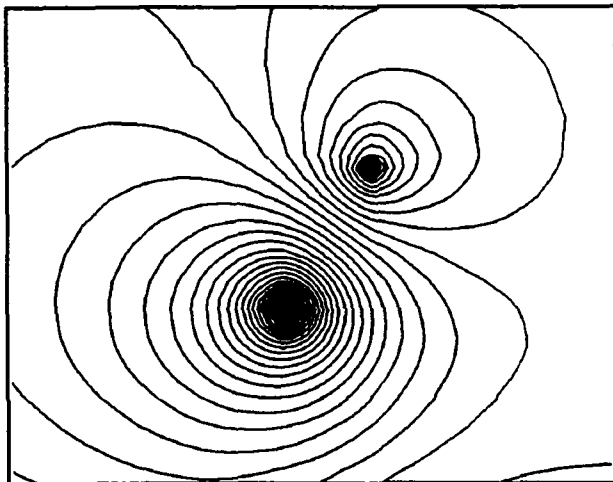


Figure 7. Model #1 Potential Distribution ( $k=1$ ).

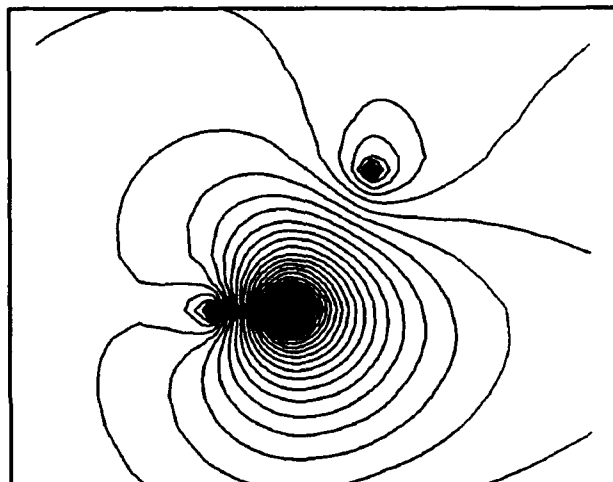


Figure 10. Model #2 Potential Distribution ( $k=1$ ).

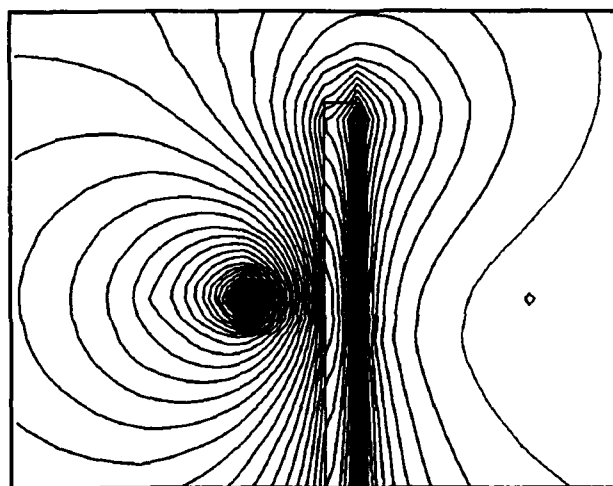


Figure 8. Model #1 Potential Distribution ( $k=17$ ).

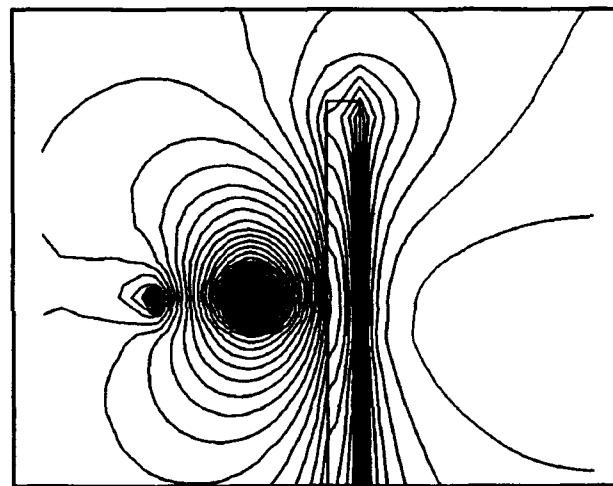


Figure 11. Model #2 Potential Distribution ( $k=17$ ).

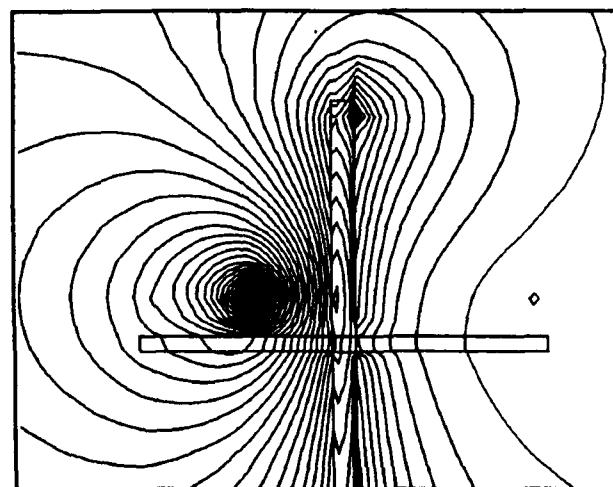


Figure 9. Model #1 Potential Distribution ( $k=19$ ).

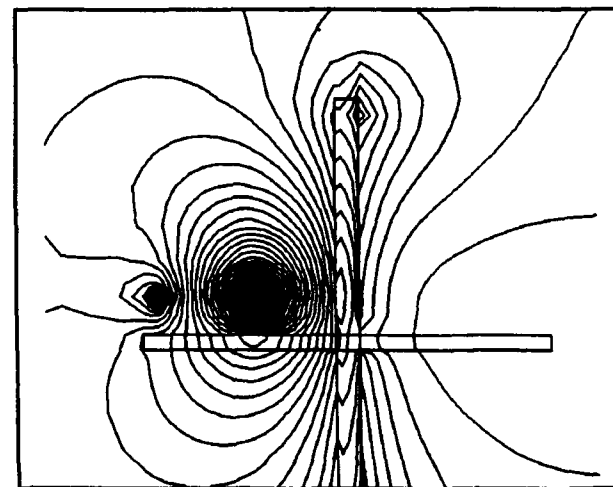


Figure 12. Model #2 Potential Distribution ( $k=19$ ).

to reduce the electric field as one moves from the surrounding air into the wood. This reduction in the electric field is caused by the bending of the equipotential contours away from the air-wood interface inside the wood region.

A more realistic model of the distribution line studied in [9] must include the insulators and associated mounting hardware. Of particular interest is the metal bolt on which the insulator is mounted. This bolt would be modelled as a perfect conductor and thus represent a floating conductor of significant cross-sectional dimension. This floating conductor would be located in close proximity to a charged conductor. The resulting equipotential contours surrounding the bolt would crowd around the equipotential volume creating a large electric field. In such a manner, the effect of the bolt would be to alter the flashover path. The present research forms the basis for further work in which the physics of the air breakdown process are incorporated into the code in an attempt to actually predict the critical flashover voltage using a computational model.

### CONCLUSIONS

The potential distributions surrounding a three-dimensional distribution line model has been computed by solving Laplace's equation throughout the enclosing volume. The partial differential equation solution was carried out using a newly developed solver code coupled with an existing grid generation (EAGLE) code. The code allows for a system model which consists of charged and/or floating conductors along with multiple dielectrics. Given a three-dimensional transmission or distribution line model, using the corresponding two-dimensional solution (the solution for the transmission line without the supporting structure) as the initial value on each cross-section of the enclosing volume enhances the convergence properties of the solution significantly. The actual potential distribution on the end faces of the enclosing volume are found to be quite similar to the corresponding two-dimensional solution.

### REFERENCES

- [1] J. D. Kraus, *Electromagnetics*, Third Edition, McGraw-Hill, 1984, pp. 63-68.
- [2] J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, *Numerical Grid Generation: Foundations and Applications*, North-Holland, 1985.
- [3] J. F. Thompson, F. C. Thames, and C. W. Mastin, "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies", *J. Comp. Phys.*, Vol. 15, No. 3, July 1974, pp. 299-319.
- [4] J. F. Thompson, "A Composite Grid Generation Code for General 3-D Regions", American Institute of Aeronautics and Astronautics 25th Aerospace Sciences Meeting, January, 1987, Reno, Nevada.
- [5] J. F. Thompson, "Program EAGLE - User's Manual, Vol. 1,2,3," USAF Armament Laboratory Technical Report AFATL-TR-88-117, Eglin AFB, Florida, 1988.
- [6] J. A. Stratton, *Electromagnetic Theory*, McGraw-Hill, 1941, pp. 163-165.
- [7] R. F. Harrington, *Time Harmonic Electromagnetic Fields*, McGraw-Hill, 1961, p.34.
- [8] E. Kuffel and W. S. Zaengl, *High-Voltage Engineering*, Pergamon Press, 1984, p. 224-228.

- [9] P. B. Jacob, S. Grzybowski and E. R. Ross, "A Guide to Estimating Lightning Insulation Levels of Overhead Distribution Lines", Mississippi State University Final Report to the Electric Power Research Institute, RP 2874-01, May 1989.

#### **ACKNOWLEDGEMENT**

Partial support for this research was provided by the Electric Power Research Institute under Research Project RP2871-1, Task 6. The remaining support was provided by the Research Center for Advanced Scientific Computing at Mississippi State University. Through funding from the National Science Foundation, the Research Center for Advanced Scientific Computing has evolved into the Engineering Research Center for Computational Field Simulation.

# COMPARISON OF ELECTROMAGNETIC RESPONSE IN TIME AND FREQUENCY DOMAINS

S. Kashyap and A. Louie  
Electronics Division  
Defence Research Establishment Ottawa  
Ottawa, Ontario, Canada  
K1A 0Z4

## ABSTRACT

This paper is concerned with the use of time- and frequency-domain methods for computing the interaction of electromagnetic waves with simple and complex structures. An example chosen for this study is a cubic box with the top open. The Finite Difference Time Domain (FDTD) method is used for computing time-domain responses to an electromagnetic pulse (EMP), a Gaussian pulse, and a sine wave. Frequency-domain results are obtained by using a moment method solution of the electric field integral equation (EFIE). Comparison is then made, both in the frequency and time domains, on corresponding quantities using Fourier transforms. Effects of various factors – the shape of the incident waveform, discretization of the structure, and Fast Fourier Transformation – on the CPU time and the accuracy of the solution are demonstrated. Guidelines are established for obtaining an accurate response.

## INTRODUCTION

Use of time-domain methods such as the FDTD for modelling a wide variety of electromagnetic interaction problems has been increasing in popularity for a number of years. Application of the FDTD method has included modelling very complex structures such as the human body, microstrip and microwave structures, radar cross-section computations and inverse scattering [1]. Response can be obtained directly in the time domain, or in the frequency domain through a fast Fourier transformation (FFT).

Frequency-domain codes such as the NEC [2] and JUNCTION [3] have also been extensively used for electromagnetic analysis of a wide variety of structures. Response obtained in the frequency domain can be converted to time domain using an inverse fast Fourier transformation (IFFT).

The choice between a frequency-domain method and a time-domain method for modelling and analyzing a specific electromagnetic interaction is not always straightforward. This paper investigates the effect of a number of factors on the accuracy of the solution obtained. These factors include incident field wave shape, structure discretization, Fast Fourier Transformation (FFT or IFFT), and computer time considerations.

## PROCEDURE

A perfectly conducting cubic box with an open top is chosen for this study. A plane wave with an EMP or a Gaussian or a sinusoidal waveform is assumed to be incident on the open face of the box. The FDTD method is used to compute time-domain fields at various points inside and outside the box for incident plane waves with different waveforms. Frequency-domain response is obtained by taking a FFT of the time-domain response. The frequency-domain responses thus obtained for various waveforms are then compared with the response obtained by using the moment method implementation of the electric field integral equation (EFIE). A de-convolution with the incident waveforms results in a waveform-independent frequency response. This results in a frequency-domain comparison.

For a time-domain comparison, the results obtained with the EFIE method are transformed into the time domain using an IFFT. A convolution with the incident waveforms results in the time-domain responses. These can then be compared with the responses obtained by using the FDTD method.

Since both the FDTD method and the EFIE method have been well described in the literature only a minimal description essential for this paper is given here. The theme of this paper is the *comparison* of results obtained from the two methods, rather than the intricacies of the methods themselves.

### a. FDTD Method:

The FDTD method is a direct implementation of the time-dependent Maxwell's equations:

$$\epsilon \frac{\partial \mathbf{E}}{\partial t} + \sigma \mathbf{E} = \nabla \times \mathbf{H} \quad (1)$$

$$\mu \frac{\partial \mathbf{H}}{\partial t} = -\nabla \times \mathbf{E}$$

The finite-difference procedure proposed by Yee [4] positioned the  $\mathbf{E}$  and  $\mathbf{H}$  fields at half-step intervals around a unit cell as shown in Figure 1, where  $\mathbf{E}$  and  $\mathbf{H}$  are evaluated at alternate half time steps, effectively giving centred difference expression for both space and time derivatives. For example, taking one of the three partial differential equations associated with each of the vector equations above gives



$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon} \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z \right) \quad (2)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left( \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right)$$

Rewriting them in finite-difference form gives:

$$E_z^{n+1}(i,j,k+1/2) = \left[ \frac{1 - \frac{\delta t \sigma(i,j,k+1/2)}{2 \epsilon(i,j,k+1/2)}}{1 + \frac{\delta t \sigma(i,j,k+1/2)}{2 \epsilon(i,j,k+1/2)}} \right] E_z^n(i,j,k+1/2) + \frac{\frac{\delta t}{\epsilon(i,j,k+1/2)}}{1 + \frac{\delta t \sigma(i,j,k+1/2)}{2 \epsilon(i,j,k+1/2)}} \left[ \frac{1}{\delta x} (H_y^{n+1/2}(i+1/2,j,k+1/2) - H_y^{n+1/2}(i-1/2,j,k+1/2)) + \frac{1}{\delta y} (H_x^{n+1/2}(i,j-1/2,k+1/2) - H_x^{n+1/2}(i,j+1/2,k+1/2)) \right] \quad (3)$$

$$H_z^{n+1/2}(i+1/2,j+1/2,k) = H_z^{n-1/2}(i+1/2,j+1/2,k) + \frac{\delta t}{\mu(i+1/2,j+1/2,k)} \left[ \frac{1}{\delta y} (E_x^n(i+1/2,j+1,k) - E_x^n(i+1/2,j,k)) + \frac{1}{\delta x} (E_y^n(i,j+1/2,k) - E_y^n(i+1,j+1/2,k)) \right]$$

where  $\epsilon$ ,  $\mu$ , and  $\sigma$  are respectively the permittivity, permeability, and conductivity of the specified coordinates in the lattice space.  $\delta x$ ,  $\delta y$ , and  $\delta z$  are the cell dimensions, and  $\delta t$  is the time between successive calculations (i.e. the time step size). For a function  $F(x,y,z,t)$  of space and time,  $F^n(i,j,k)$  is Yee's notation for the value  $F(i\delta x, j\delta y, k\delta z, n\delta t)$ .

The complete system of six finite-difference equations then provides a computational scheme: the new value of a field vector component at any point depends only on its previous value and on the previous values of the components of the other field vector at adjacent points. Thus at any given time step the computation can proceed one point at a time for a single processor or several points at a time for a machine with parallel processors.

While not the subject of this paper, the following comment on the FDTD algorithm may be of interest. The finite-difference form (3) is obtained from (2) by the approximation

$$\sigma E(x,y,z,t) \approx \sigma \left( \frac{E^{n+1}(i,j,k) + E^n(i,j,k)}{2} \right) \quad (4)$$

This approximation is used in many FDTD studies (e.g. [1]). Others (e.g. [5], [6]) have used the approximation

$$\sigma E(x,y,z,t) \approx \sigma E^n(i,j,k) \quad (5)$$

and obtained good results. The approximation (5), however, may lead to instability if

$$\sigma \delta t / \epsilon > 1 \quad (6)$$

In this work, the approximation (4) is used. But even if we had used approximation (5), because of our special treatment of boundaries for perfectly conducting bodies we would still have had stable results. For a perfectly conducting body, we have a boundary-checking algorithm that selects the boundary faces on which to set the tangential  $E$ -fields to zero. This boundary is thus a "sharp" one of zero thickness and not a "fuzzy" one-cell-thick wall with a huge  $\sigma$ . For a dielectric surface, we use a "harmonic mean" method to smooth out the boundary transitional effect. Another necessary key for stability of the time-stepping algorithm (3) is that the time step  $\delta t$  is chosen to satisfy

$$c \delta t \leq \left( \frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2} \right)^{-1/2} \quad (7)$$

#### b. EFIE Method:

Reference [7] describes a simple and efficient numerical procedure for scattering by arbitrarily shaped bodies, using the moment method to solve the electric field integral equation (EFIE). The object surface is modelled by using planar triangular patches (for example, Figure 2). Because of the EFIE formulation the procedure is applicable to both open and closed surfaces. The procedure has been applied to a wide variety of electromagnetic interaction problems and has yielded excellent correspondence between the exact formulations and other methods. In JUNCTION, the EFIE approach is extended to analyze an arbitrary configuration of conducting wires and bodies. The algorithm developed can handle wire-to-wire, surface-to-surface and wire-to-surface junctions. A modified version of JUNCTION is used here as the "EFIE method".

## PLANE WAVE FORMS

In this study, the time-domain incident wave on a structure is a plane wave with one of the following three shapes:

a. the nuclear electromagnetic pulse (NEMP) [8]

$$E(t) = E_0 \frac{e^{\alpha t}}{1 + e^{(\alpha+\beta)t}} \quad (8)$$

with  $E_0 = 5.126 \times 10^4 \text{ V m}^{-1}$ ,  $\alpha = 1.027 \times 10^9 \text{ s}^{-1}$ , and  $\beta = 3.906 \times 10^6 \text{ s}^{-1}$  (see Figure 3a). [Note this pulse has a peak value of 50 kilovolts per metre at 10 nanoseconds, a 10 to 90 percent rise time of 5 nanoseconds, and a decay time to half-value of 200 nanoseconds.]

b. the Gaussian pulse

$$E(t) = E_0 e^{-\pi \left(\frac{t}{T}\right)^2} \quad (9)$$

with  $E_0 = 100 \text{ V m}^{-1}$ , and  $T = \sqrt{\pi} (m \cdot \delta t)$ , where  $\delta t$  is the time step size and  $m$  is the "pulse width" parameter: when  $t = m \cdot \delta t$ ,  $E(t) = E_0 / e \approx 0.37 \cdot E_0$ . Figure 3b shows two different pulse widths.

c. the sine wave

$$E(t) = E_0 \sin\left(\frac{2\pi t}{N \cdot \delta t}\right) \quad (10)$$

with  $E_0 = 1 \text{ V m}^{-1}$ , and frequency  $f_0 = 1/(N \cdot \delta t)$  [  $\omega_0 = 2\pi/(N \cdot \delta t)$  ], where  $N$  is the number of time steps of  $\delta t$  each, whence  $N \cdot \delta t$  is the period. Figure 3c shows ten cycles of the sine pulse with a period of 5 ns, hence a frequency of 200 MHz.

Note the different abscissa and ordinate scales used in Figures 3a-3c. These three waveforms, when fast-Fourier transformed into the frequency domain, have the frequency spectra shown in Figures 4a-4c.

For the NEMP, note that the frequency spectrum reaches 1% of its peak value at about 100 MHz, 0.1% at about 220 MHz, and 0.01% at about 330 MHz. From 400 MHz

on numerical noise enters into the FFT process.

For the Gaussian pulse, note that narrower time-pulses have wider frequency spectra, and that 1% and 0.1% of peak values in frequency spectra are reached in the Gigahertz range (in our example where  $\delta t \approx 4 \times 10^{-11}$  s). For the  $m = 12$  case, 1% is reached at about 1.5 GHz and 0.1% at about 1.8 GHz, and numerical noise dominates (i.e. the real signal falls below the "noise floor" value of about  $10^{-15}$ ) after 2.5 GHz. For the  $m = 6$  case these "break points" are about doubled.

The theoretical frequency spectrum of the sine pulse is, of course, the delta function centred at  $f_0$ . Figure 4c shows the FFT representation of  $\delta(f - f_0)$ , i.e. the sinc function. (Figure 4c is the only one among 4a-4c that shows a "truncation effect". In figures 3a and 3b, the time-domain values of the pulses are taken until the pulses have "gone through", i.e. until the pulse values are negligible, so the frequency spectra in Figures 4a and 4b are "complete". This is, of course, not possible in the sine wave 3c.)

### FDTD RESULTS FOR AN OPEN BOX

We use, as the example in this study, a perfectly conducting cubic box with an open top, and an incident  $x$ -polarized plane wave propagating in the  $-z$  direction. Each edge of the cubic box is 30 cm, and the  $x$ ,  $y$ , and  $z$  coordinates range from 0 to 0.3m. The cubic box is divided into  $13 \times 13 \times 13$  Yee cells, centrally located within an FDTD cell space of  $60 \times 60 \times 60$  cells. Four field points are chosen for comparison between their time- and frequency-domain  $E_x$ -field responses. These points are labelled A,B,C,D and are at a distance of 0.0577m from the " $x=0$ "-wall, 0.1385m from the " $y=0$ "-wall, and 0.0923, 0.2077, 0.3000 (at the "mouth" of the box), and 0.5077 ("outside" the box) metres from the bottom ( $z=0$ ) of the box, respectively. Since each cubic Yee cell has an edge length of  $0.3\text{m}/13 = 0.0231\text{m}$ , these field points are 2.5 space steps from the "back", 6 space steps from the "side", and, respectively, 4, 9, 13, and 22 space steps from the bottom. [The  $x$ -coordinates are half a space step off because in the Yee cell, the  $E_x$ -field component is evaluated at  $(x + \delta x, y, z)$ .] Figure 5 shows the boundary faces of this open box, on which the tangential  $E$ -fields are set to zero.

Figures 6-8 show the time-domain  $E_x$ -field response, at the selected field points, to incident NEMP, Gaussian, and sine pulses.

Figures 9-10 show the frequency-domain  $E_x$ -field response at the selected field points, obtained from a fast Fourier transformation with de-convolution of the incident pulse of the corresponding time-domain curves in Figures 6 and 7. For the responses to the NEMP in Figure 6, since it was too time-consuming to run FDTD for enough time steps for them to decay down to close-to-zero values, they are extrapolated for later time

using a simple exponential decay curve. (It is necessary in the time-to-frequency Fourier transform for the time-domain function values to reach close to zero for the Fourier integral not to be "truncated".) The magnitude of the frequency response (in this case at 200 MHz) corresponding to the sine pulse in Figure 8 are simply the stationary time-domain response peak values (also shown in Figure 8).

The corresponding curves in Figures 9 and 10 compare reasonably in an overall qualitative way. The excessive "wiggling" of the curves in Figure 9 beyond 300 MHz is due to the numerical noise in the FFT-frequency spectrum of the NEMP curve for higher frequencies (as noted above). Thus the results in Figure 9 are only reliable up to about 300 MHz. Because for the Gaussian pulse (with  $m=12$ ) numerical noise does not set in until after 2.5 GHz, we may be tempted to "trust" the results in Figure 10 for the whole domain (up to 1.75 GHz) shown. There is, however, another limitation in force. The spatial resolution of the FDTD box is  $0.0231\text{m}$ , and so for reasonable accuracy the minimum wavelength should be  $10 \times 0.0231 = 0.231\text{m}$ , whence the maximum frequency is 1.3 GHz.

Thus, in the domain 0-300 MHz, the corresponding curves in Figures 9 and 10 are identical. Also, the frequency responses obtained from the time-domain incident sine wave in Figure 8 match these curves at 200 MHz. We may therefore conclude that any one of the incident waves may be used to run FDTD, and *within the numerically reliable part of their frequency spectra*, the fast-Fourier transformed response in frequency domain are comparable. From a computational-time standpoint, it is therefore more efficient to run FDTD with the Gaussian pulse, as less time steps are needed for completion (i.e. for the response fields to decay to close-to-zero values).

## EFIE RESULTS AND COMPARISON IN FREQUENCY DOMAIN

For the EFIE method in the frequency domain, the same open-topped box is used, subjected to an incident  $E_x$ -polarized plane wave travelling in the  $-z$  direction at various frequencies. Several geometric versions of the box are used, representing various resolution requirements: recall the one-fifth wavelength rule, that the maximum edge length on the structure must be at most one-fifth of the incident wave length for the field results to have reasonable accuracy. Two versions are shown in Figure 11.

There are two different ways to represent the frequency-domain response field data. One way is for a fixed field point, EFIE is run for a whole domain of different frequencies (e.g. every 10 MHz step up to 1.6 GHz), and the resulting  $E$ -field versus frequency data set is directly comparable to the Fourier-transformed data from FDTD such as those shown in Figures 9 and 10. A second, more common, way is for a fixed frequency, EFIE is run for a set of field points (e.g. for the box at  $(0.0577, 0.1385, z)$  where  $z$  ranges from  $-0.1$  to  $0.6$  with  $\delta z = 0.01$ ). To compare the  $E$ -field versus location data set with FDTD, the time-domain FDTD response at many field points are taken and then

Fourier-transformed to frequency domain, and one value at the particular frequency for each field point is collected.

As an example of the first type of comparison, consider the field points  $C = (0.0577, 0.1385, 0.3000)$  and  $D = (0.0577, 0.1385, 0.5077)$ . For each field point, EFIE is run for every 10 MHz, from 10 MHz to 1.6 GHz, and the  $E_x$ -field values at C and D are evaluated. The results are the solid curves shown in Figure 12. The dashed curves are from FDTD, viz the curves in Figures 10c and 10d. The comparison is reasonably good, and we shall discuss the discrepancies (especially above 1 GHz) in a later section. The comparison differs from point to point and is better at D (and at other points) than at C. We shall use the "worst" point C among the four and the "good" point D for further illustration and analysis.

As an example of the second type of comparison, consider at a fixed frequency 200 MHz, the set of field points  $\{(0.0577, 0.1385, z): -0.1 \leq z \leq 0.6, \text{ with } \delta z = 0.01\}$ . (Note that this field line passes through the points A-D.) EFIE is run at 200 MHz, and the  $E_x$ -field values at these points are evaluated. The result is the curve shown in Figure 13. FDTD, on the other hand, is run with (arbitrarily) nine field evaluation points. The resulting time-domain  $E_x$ -field data are then transformed to frequency domain, and  $E_x(f)$  at  $f = 200$  MHz at these nine points are the circles in Figure 13. It makes no significant difference in this case (i.e. at this frequency) which incident wave is used in FDTD, as we observed in the previous section. Again, the comparison is reasonable, and the minor differences will be discussed later.

### COMPARISON IN TIME DOMAIN

Comparison between FDTD and EFIE can also be made in time domain. When the  $E_x$ -field versus frequency EFIE curves of Figure 12 are inverse-Fourier transformed to time domain and convolved with the Gaussian pulse, we obtain the solid curves in Figure 14, which are almost identical to the FDTD results of Figures 7c and 7d (shown as the dashed curves in Figure 14).

When the EFIE curves are inverse-Fourier transformed to time domain and convolved with the NEMP, however, we obtain the solid curves in Figure 15. At the field point D the solid curve compares well with the dashed curve, which is the FDTD result of Figure 6d. But at the field point C, the solid curve is significantly different from the dashed curve, which is the FDTD result of Figures 6c.

The key to the explanation of this apparent difficulty in time domain comparison, at the field point C when the incident plane wave is the NEMP, is in the width of the frequency spectrum. For the Gaussian pulse with  $m = 12$ , over the EFIE domain in Figure 12 from 0 to 1.6 GHz, the frequency spectrum just decreases from its peak value to 0.1%

(see Figure 4b). Thus the whole data set is significant in the inverse-Fourier transform. And since the solid curves and the dashed curves in Figure 12 are relatively similar, their transforms into time domain in Figure 14 are also similar. (The inverse-Fourier transform of the dashed curves in Figure 12 – i.e. of the curves in Figures 10c and 10d – are of course just Figures 7c and 7d.) For the NEMP, 0.1% of the peak value is already reached at about 220 MHz, hence only the low-frequency portion of the curves in Figure 12 are significant in the inverse-Fourier transforms. (The FDTD curves that should be used here are actually Figures 9c and 9d, but in the domain from 0 to 220 MHz Figures 9c and 10c, and Figures 9d and 10d – the dashed curves in Figure 12 – are identical.) Observe that for the field point D in Figure 12, from 0 to 220 MHz, the solid and dashed curves are very similar, and so their transforms into time domain in Figure 15 are also very similar. But for the field point C, in the domain from 0 to about 150 MHz in Figure 12, the two curves are very different, and so their transforms into time domain in Figure 15 are different. (The inverse-Fourier transforms of Figures 9c and 9d are Figures 6c and 6d, respectively.)

So the question becomes: why, as in Figure 12 for the field point C, does the frequency domain comparison not fare well for low frequencies ( $< 150$  MHz)? Here is where the different geometric versions of the patch-model box become a factor – but not in the expected way due to the one-fifth wavelength rule.

All the EFIE results presented so far are done with the coarser box in Figure 11, i.e. the one where the edge of the cube is divided into four equal parts. In this model, the maximum-length edges are the diagonals, which are  $\sqrt{2} \times 0.3/4 = 0.106$  m. Hence by the one-fifth wavelength rule this box is good for frequencies up to about 566 MHz. The other box in Figure 11 has the cubic edge divided into ten equal parts, and by the same rule is good up to about 1.4 GHz. The one-fifth wavelength rule, however, sets a limitation on *high* frequencies, and so does *not* explain the low frequency difficulties. In fact, the one-fifth wavelength rule may be more stringent than what is observed in practice. We could run the coarser box up to 1.6 GHz and the results obtained up to about 1 GHz are very similar to those from the finer box. The use of the coarser box is the reason why in Figure 12 the two curves do not match well above 1 GHz.

As it turns out, however, the box with the finer grid does give better *low-frequency* values. The reason is that in the calculation of near fields from the currents on the edges, there must be a fine enough spatial resolution in the geometric structure to reflect the highly varying field values, especially when close to boundary edges (i.e. those edges around the opened top). This "edge effect" seems to be more pronounced at low frequencies. The numbers on the dashed curves in Figure 16 represent the number of divisions of the cubic edge into equal parts. Note that there is no significant improvement in using a finer division than edge/10. Figure 17 shows the corresponding comparison in time domain. There is still significant difference between IFFT(EFIE) and FDTD.

## FDTD DISCRETIZATION

One limiting feature of the Yee-cell FDTD formulation is that the various components of the electric and magnetic fields are assumed to be constant within each Yee cell, thus these field values are "discretized" in spatial steps. We have been evaluating the  $E_x$ -field component at  $(x+\delta x/2, y, z)$  in EFIE, because these are the coordinates where the Yee-cell  $E_x$ -fields are attached. When we try field points with EFIE in close neighbourhoods around the points  $(x+\delta x/2, y, z)$ , however, we manage to get good comparison between IFFT(EFIE)\*NEMP and FDTD. Here EFIE is run with the edge/10 finer grid model of the box.

For example, around a neighbourhood of the field point  $C = (0.0577, 0.1385, 0.3)$ , we find that evaluating the EFIE  $E_x$ -field at  $C' = (0.0577, 0.1385, 0.2850)$  gives the best match between EFIE and FDTD in time domain. Moving the point  $C'$  slightly in the  $x$  direction yields minor variations in the  $E_x$ -field, moving slightly in the  $y$  direction yields no change, while moving in the  $z$  direction yields the most significant changes. The best match is when  $C' = C - (0., 0., 0.0150)$ . See Figure 18. (We have only tried varying one spatial direction at a time for simplicity. It is entirely possible that the best match in fact occurs at a point where all three coordinates differ slightly from  $C$ .)

Similarly, for the other three field points, we find the best matches at  $A' = A + (0., 0., 0.0090)$ ,  $B' = B - (0., 0., 0.0140)$ , and  $D' = D$ . Field point  $C$  requires the largest spatial shift for comparison because around the "mouth" of the box, the field values have the largest variations with respect to position.

Thus, the corresponding field evaluation points that give the best match in time-domain between EFIE and FDTD are within a spatial step  $\delta z$  in the  $z$  direction of each other (i.e. within the same Yee cell). This is accountable as "discretization error", as the FDTD fields are discrete approximations of the "smooth" EFIE fields. Figure 19 shows the origin of this discretization error. The "central differencing scheme" of the FDTD approach approximates the derivative of a smooth function  $f(x)$  at a point  $a$  by

$$\frac{df}{dx}(a) \approx \frac{\delta f}{\delta x}(a) = \frac{f\left(a+\frac{h}{2}\right) - f\left(a-\frac{h}{2}\right)}{h} \quad (11)$$

where  $h$  is the differencing interval. But the value of this "approximate derivative" is not necessarily the exact value of  $df/dx$  at  $a$ . The mean value theorem for derivatives in elementary calculus only guarantees the existence of a value  $a'$  somewhere between  $a-h/2$  and  $a+h/2$  with



$$\frac{df}{dx}(a') = \frac{\delta f}{\delta x}(a) \quad (12)$$

This is why the exact match between EFIE and FDTD occurs not necessarily at the same field point but within a spatial step.

(An alternative hypothesis exists for the non-correspondence between the FDTD field point and the EFIE field point: in [9] it is stated that the discrepancy may be due to FDTD's spatial approximation at the box surface. But this "fuzzy boundary" is an artifact of using an FDTD body with a one-cell-thick wall. Our version of the FDTD model for a perfectly conducting body, as we mentioned before, has a "sharp" boundary of zero thickness. The uncertainty in the distance of the field point from the surface of the body is, therefore, not an issue in our algorithm.)

Theoretically, therefore, if  $h$  is made smaller, the difference between  $a$  and  $a'$  may become smaller. That is to say, that if FDTD is run with smaller cells (finer resolution), the spatial difference between matching FDTD and EFIE field points will be smaller. But using *smaller* cells also means *more* cells, and then computer memory and running time become factors.

### CPU-TIME CONSIDERATIONS AND MODELLING GUIDELINES

We have shown that in computer simulations of the interaction of electromagnetic waves with geometric structures, both time- and frequency-domain codes may be used. The two independent methods are comparable – as long as proper precautions are taken – and can be used as verification of the accuracy of each other.

From an efficiency, i.e. CPU-time economy, point of view, the FDTD method with an incident Gaussian pulse is the approach of choice. For the open box example, running EFIE takes about 3 hours of CPU-time on a VAX 6420 *for each frequency*, running FDTD with the Gaussian pulse (2000 time steps) takes about 6 hours, and running FDTD with the NEMP (10000 time steps) takes about 30 hours (and the latter still needs further extrapolation). Other geometric structures also have a similar CPU-time ratio, that the CPU-time taken for EFIE(one frequency) :: FDTD(Gaussian) :: FDTD(NEMP) is 1::2::10.

The reason that FDTD(Gaussian) is the most efficient is that the time-domain response decays back to zero rapidly, and that after a complete run, one can Fourier-transform the results (with de-convolution of the Gaussian pulse) and obtain the field response *for all frequencies* (within the wide frequency spectrum of the Gaussian pulse). In other words, in the time it takes EFIE to run two frequencies, the process

$$\text{FFT/Gaussian [ FDTD(Gaussian) ]} = \text{EFIE(all frequencies)}$$

gives the whole frequency spectrum of responses. Because frequency-domain response comparison, with FFT(FDTD) versus EFIE, has been shown to be reasonably accurate, this process is a reliable and time-saving method in obtaining frequency-domain data.

In time domain, if one simply wants the *early-time* response to the NEMP, one may run FDTD(NEMP) directly. If, however, one is in fact interested in the *late-time* EMP response, one can run FDTD(Gaussian), then Fourier-transform to frequency domain with de-convolution of the driving Gaussian pulse, and then inverse-Fourier-transform the frequency-domain response thus obtained and convolve with the NEMP; i.e. through the process

$$\text{IFFT [ FFT/Gaussian [ FDTD(Gaussian) ] ]} * \text{NEMP} = \text{FDTD(NEMP)}.$$

This way, FDTD only has to be run for the small number of time steps that an incident Gaussian pulse requires, instead of the long duration of the NEMP pulse. Several EFIE runs at selected frequencies and a direct FDTD(NEMP) run (for a smaller number of time steps) can always be used as checks to insure accuracy of this approach.

Thus, in summary, the merits of the FDTD method with an incident Gaussian pulse, followed by a time-to-frequency Fourier transform, are:

- a. large frequency content of the incident pulse,
- b. pulse decays down to zero rapidly, minimizing running time, and
- c. efficiency: one time run to obtain all frequencies.

(Note, however, there is nothing "magical" about the Gaussian pulse itself: any time-domain pulse of narrow pulse width would share the same merits. The Gaussian pulse is chosen because of its simple analytic form and because it is a "standard".) The main disadvantage is due to computer resources, that only the chosen field quantities at several specified points are written to the output (although all six field components at all the Yee cells are evaluated at each time step, due to the constraint of the size of the output file only those chosen ones are written out). The code must be run again for computation of other field components and at other points. (As a contrast, in EFIE the currents on all the edges are stored in an output file. So the field values at any other points *at the same frequency* can be calculated from this "currents file" and EFIE does not have to be rerun.)

Time-domain response comparison has some inherent inaccuracies, mainly due to the fact that difference equations are by definition *approximations* to differential equations. In FDTD versus IFFT(EFIE), care has to be taken in finding the correct field locations for direct comparisons. Frequency-to-time inverse Fourier transformation also has some inherent problems. For a complete time-domain response it is less efficient from CPU-time considerations as described before. In addition, even for early-time response determination one still has to calculate the frequency response at a large number of frequencies to obtain an accurate IFFT into time-domain.

Finally, it must be remembered that discretization errors can be significant. In the FDTD approach one must keep in mind that the minimum reliable wavelength is ten times the size of the Yee cell (hence setting the limit for the maximum reliable frequency). Also, using smaller cells (hence more cells), within the limit of the host computer, to model the geometric object may improve the accuracy of the comparison. The availability of the field quantities only at discrete points due to the lattice structure can create some problems. In the frequency-domain code EFIE, discretization affects both the high and the low frequencies: on the one hand there is the one-fifth wavelength rule we discussed, setting the limit for the maximum frequency, and on the other hand at low frequencies there must be enough spatial resolution to reflect highly varying fields in neighbourhoods of "boundary edges". It must be remembered that the discretization guidelines of "10 cells/ $\lambda$ " and "edges  $\leq \lambda/5$ " are "traditional" ones based on experience from many studies in computational electromagnetics. They are sometimes more stringent than necessary and useful results may be obtained even above the high-frequency threshold. This is why in some of our figures (notably Figure 9) we have presented the high-frequency results well above the threshold. The point of caution is that if the guidelines are violated, one must seek independent verification of the results obtained.

## CONCLUSIONS

In this paper, the penetration of electromagnetic waves inside an open-topped cubic box has been studied. The FDTD code has been used to calculate the time-domain response for an EMP, a Gaussian pulse, and a sine wave. Comparison, in both time and frequency domains, has been made with the results obtained by using the frequency-domain method EFIE. Effects of various factors such as wave shape, structure discretization, and fast Fourier transformation on CPU-time and accuracy of the results were discussed. Guidelines for using the time-domain and the frequency-domain codes were suggested. It was found to be more efficient in most cases to use the time-domain method.

*Acknowledgement:* We would like to thank the *ACES Journal* editor and referees for their constructive comments.

## REFERENCES

- [1] A. Taflov and K. R. Umashankar, "The Finite-Difference Time-Domain Method for Numerical Modeling of Electromagnetic Wave Interactions with Arbitrary Structures", *Progress in Electromagnetic Research*, Elsevier, 1990.
- [2] G. L. Burke and A. J. Poggio, "Numerical Electric Code (NEC) – Method of moments", *Naval Ocean Systems Centre, Technical Document 116*, 1977.

- [3] S.-U. Hwu and D. R. Wilton, "Electromagnetic Scattering and Radiation by Arbitrary Configurations of Conducting Bodies and Wires", *University of Houston, Technical Report 87-17*, 1988.
- [4] K. S. Yee, "Numerical Solution of Initial Boundary Value Problem involving Maxwell's Equations in Isotropic Media", *IEEE Transactions on Antennas and Propagation*, AP-14, 302-307, 1966.
- [5] A. Taflov and M. E. Brodwin, "Numerical Solution of Steady-State Electromagnetic Scattering Problems Using the Time-Dependent Maxwell's Equations", *IEEE Transactions on Microwave Theory and Techniques*, MTT-23, 623-630, 1975.
- [6] J. Van Hese and D. De Zutter, "Modeling of Discontinuities in General Coaxial Waveguide Structures by the FDTD-Method", *IEEE Transactions on Microwave Theory and Techniques*, MTT-40, 547-556, 1992.
- [7] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic Scattering by Surfaces of Arbitrary Shape", *IEEE Transactions on Antennas and Propagation*, AP-30, 409-418, 1982.
- [8] "EMP Engineering Handbook", *NATO File No. 1460-3*, November, 1989.
- [9] C. Trueman, S. Kubina, R. Luebbers, K. Kunz, S. Mishra, and C. Larose, "Validation of FDTD RCS Computations for PEC Targets", *Proceedings of the IEEE Antennas and Propagation Society International Symposium, Chicago, IL, July 18-25, 1992*.

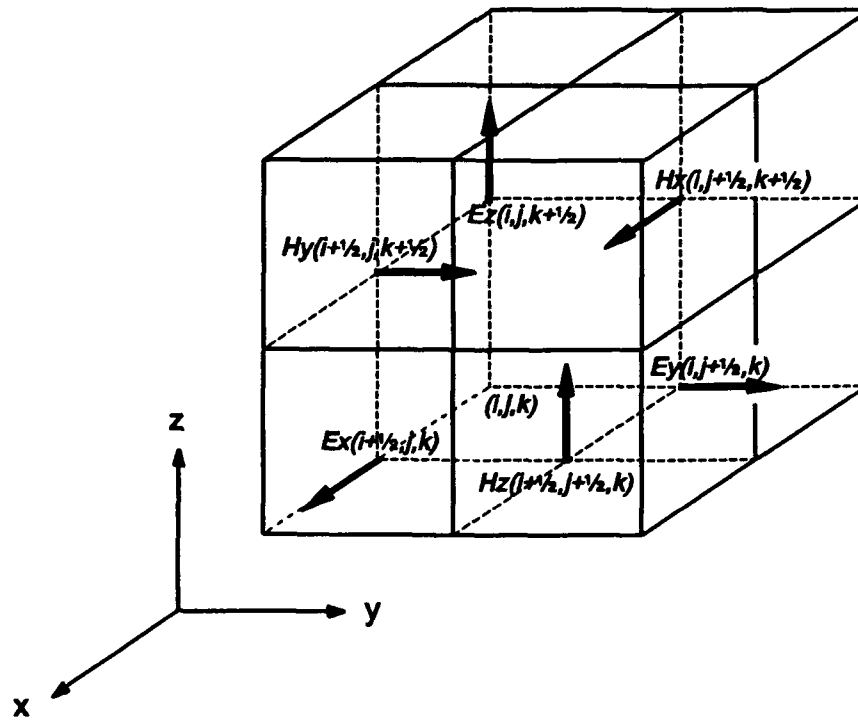


Figure 1. Position of the field components in a unit cell of the Yee lattice.

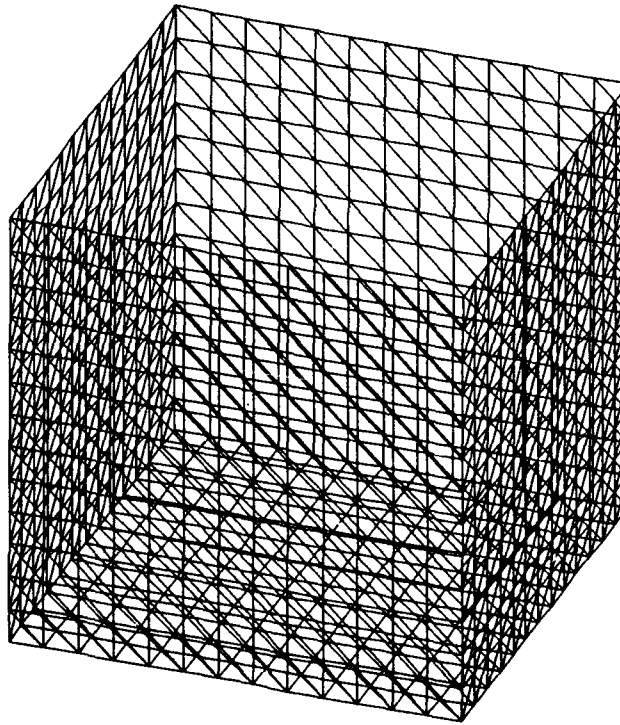


Figure 2. Example of a triangular surface-patch model input file for EFIE.

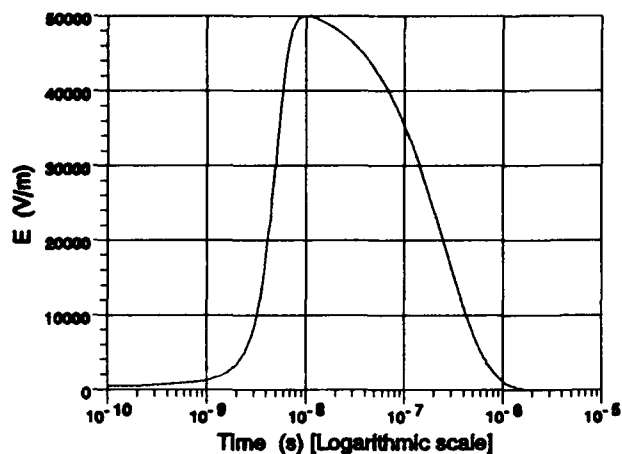


Figure 3a. NEMP in time domain

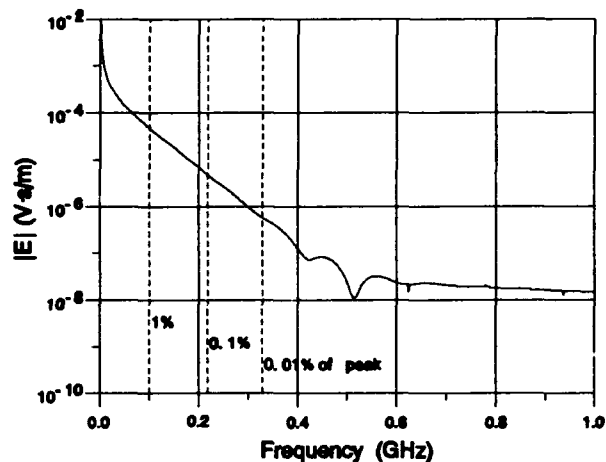


Figure 4a. NEMP in frequency domain

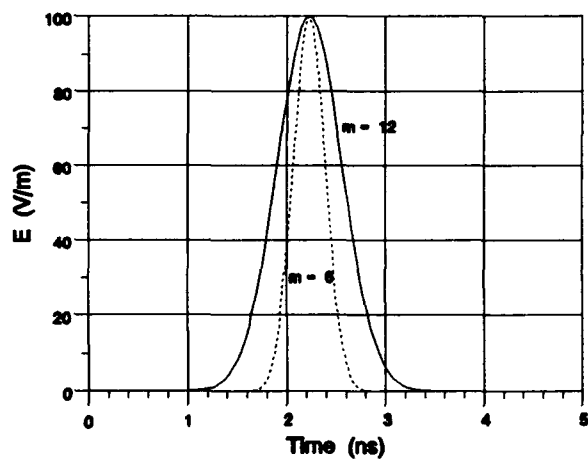


Figure 3b. Gaussian pulses with different pulse widths, in time domain

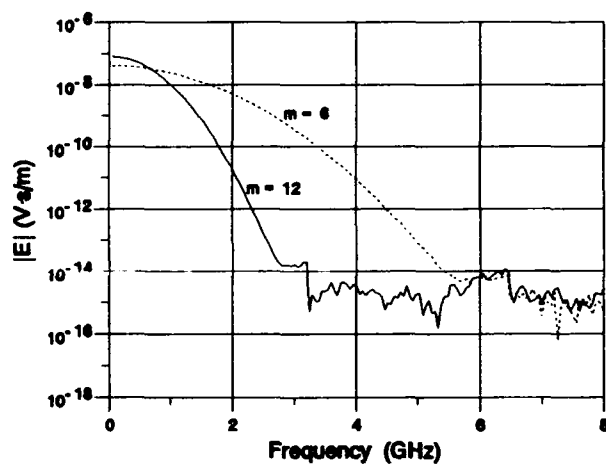


Figure 4b. Gaussian pulse with different pulse widths, in frequency domain

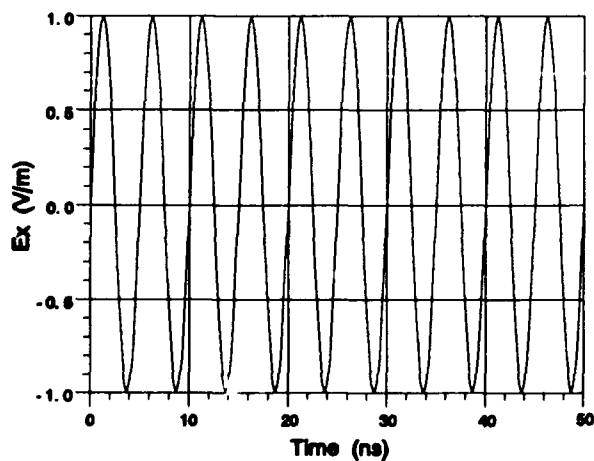


Figure 3c. Sine pulse in time domain, period = 5 ns

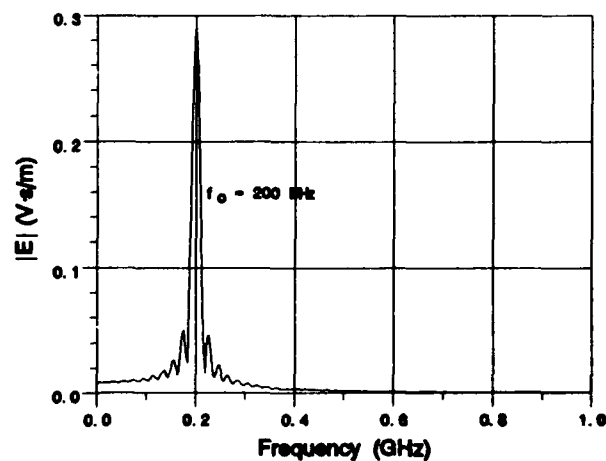


Figure 4c. Sine pulse in frequency domain, frequency = 200 MHz

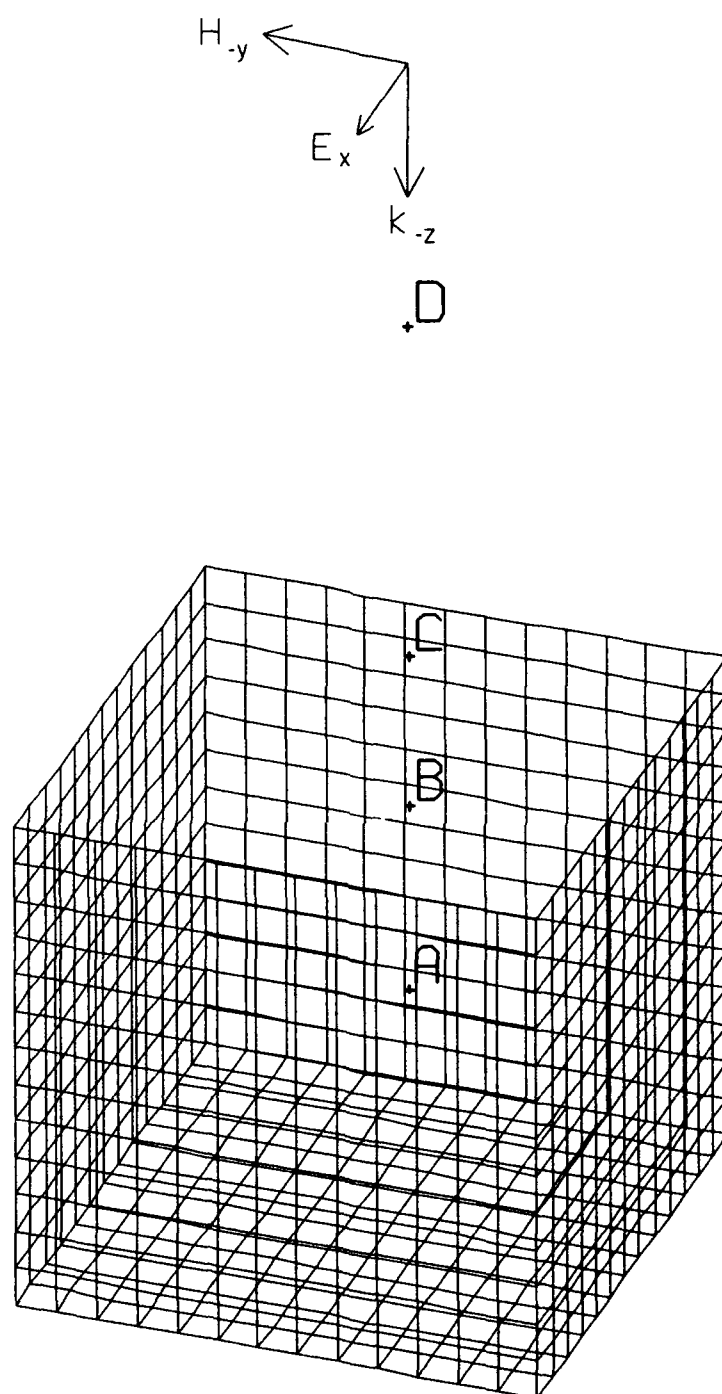


Figure 5. The open-topped box used in FDTD studies

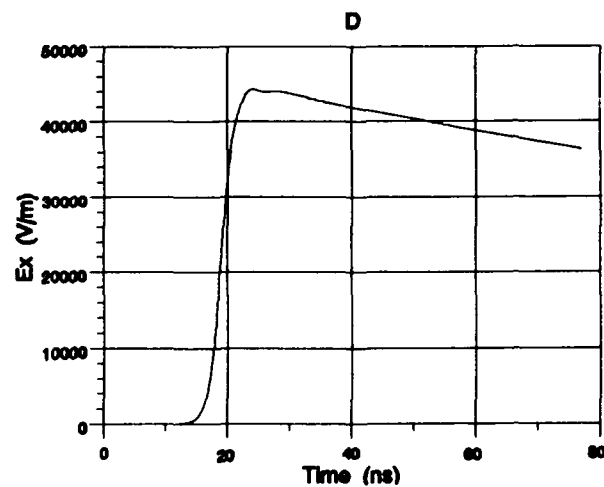
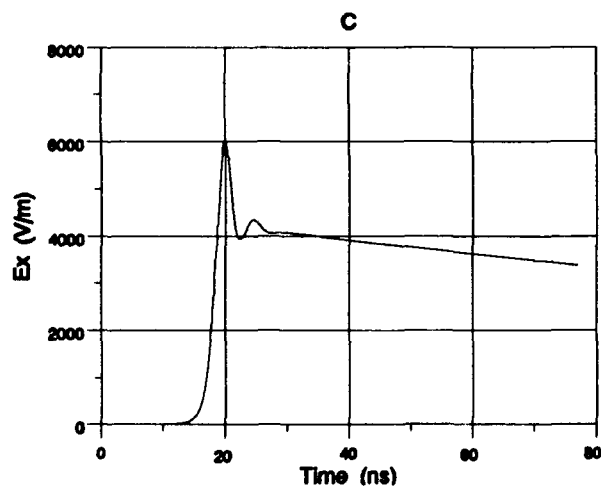
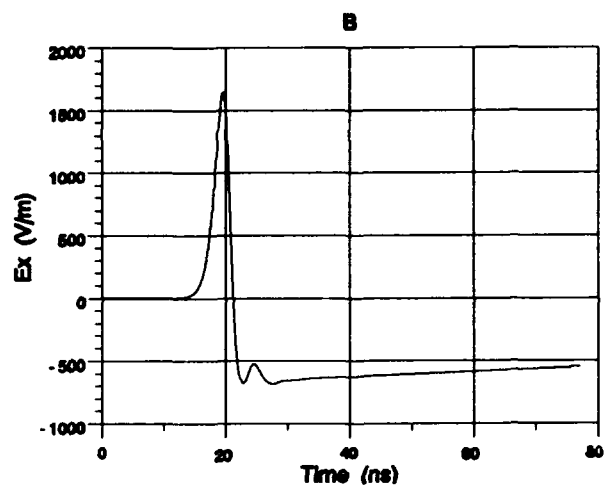
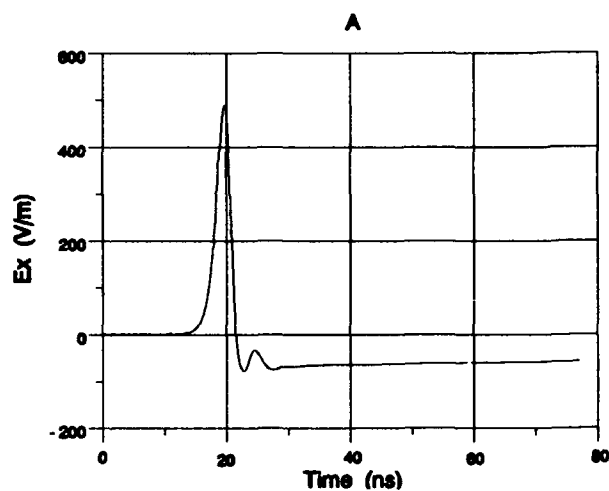


Figure 6. The Ex-field time-domain response at the four field points:  
...to the NEMP,...



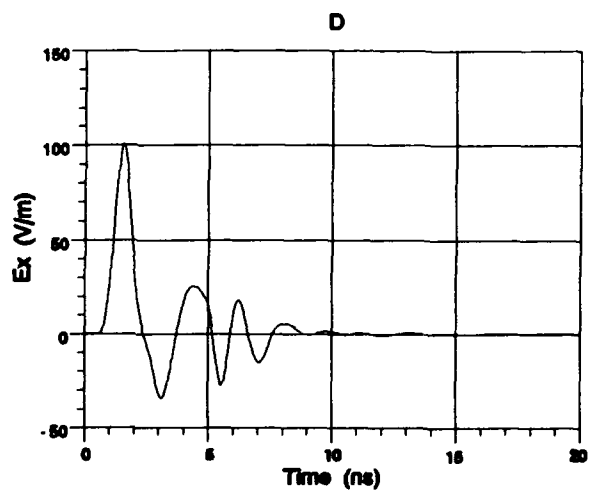
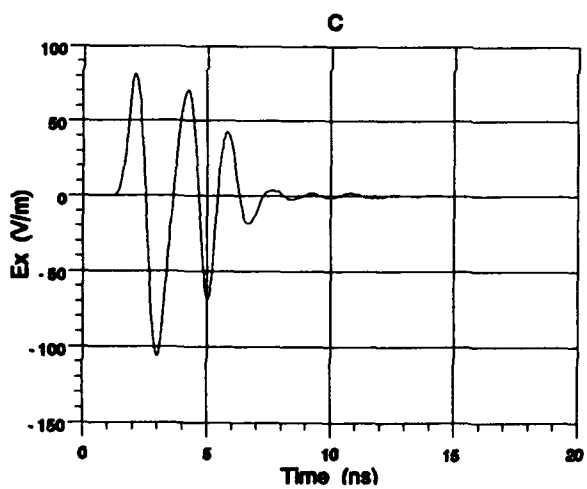
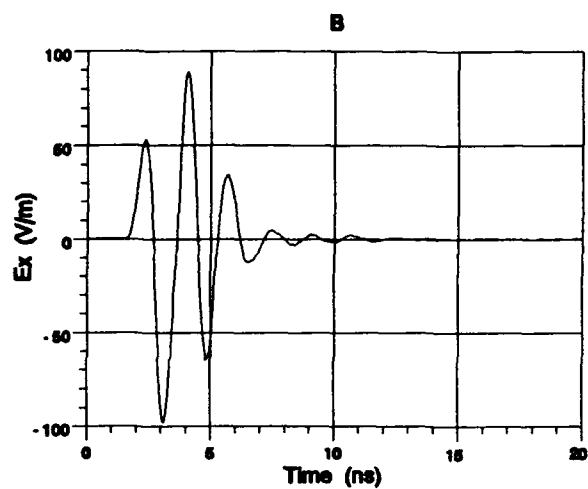
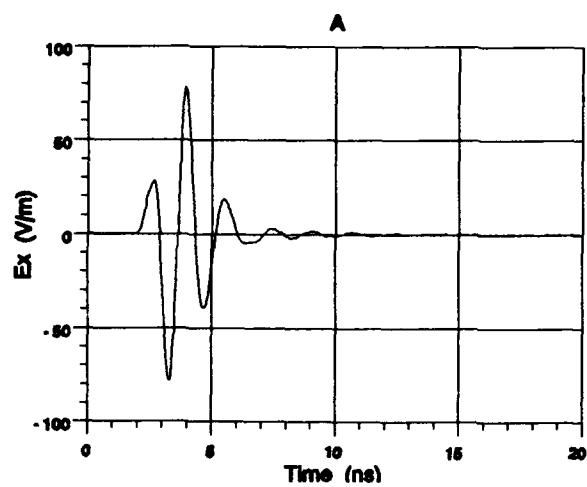


Figure 7. ...to the Gaussian pulse with  $m=12$ ,...

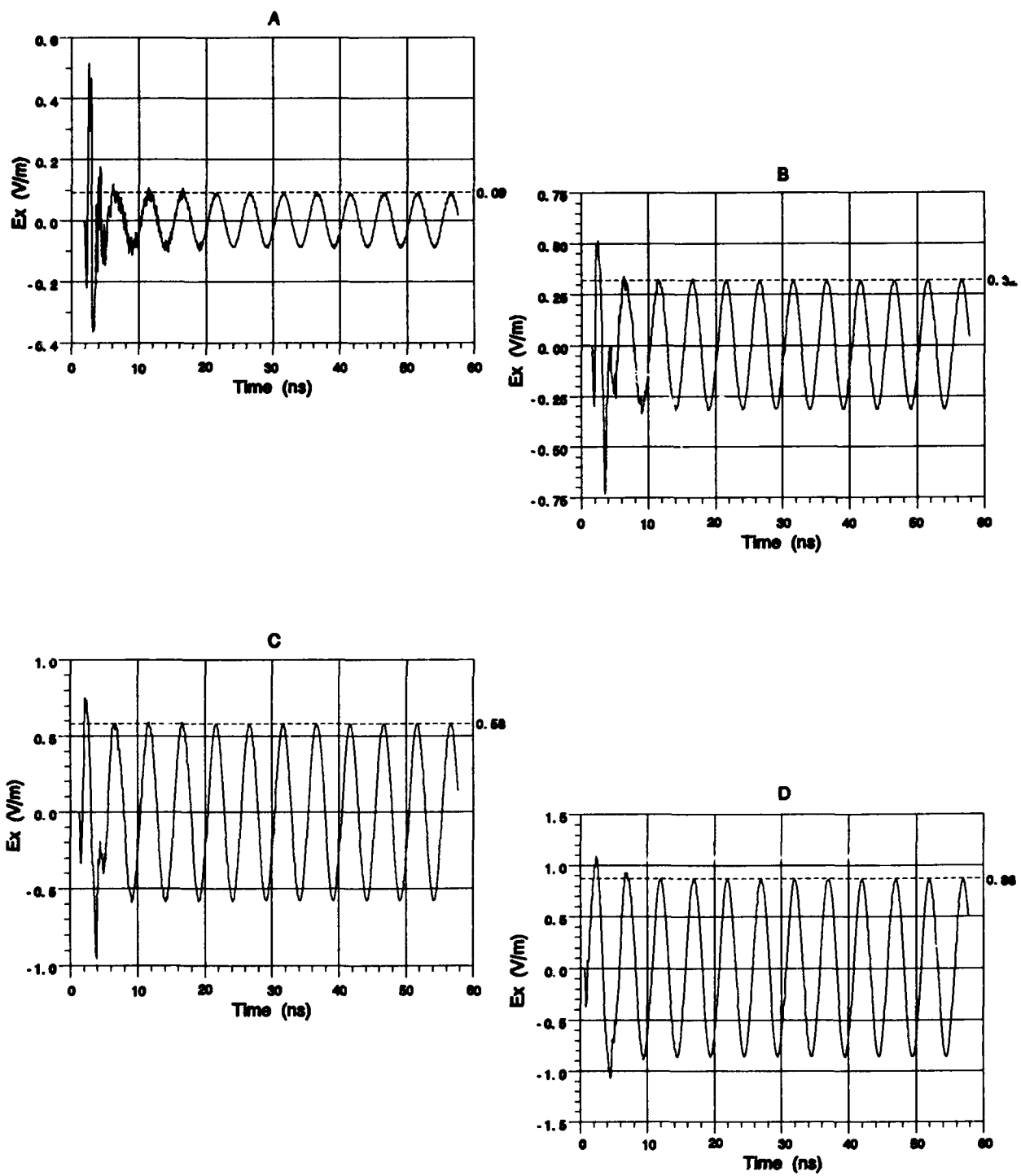


Figure 8. ...and to the sine pulse at 200 MHz.

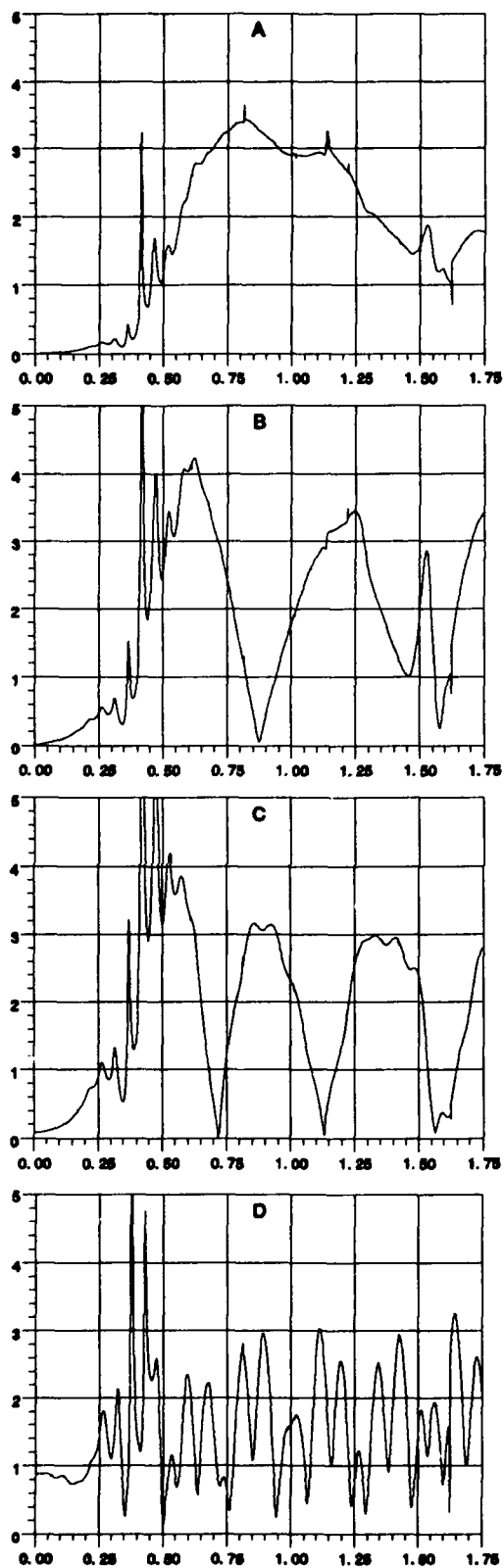


Figure 9. Fourier transforms of the curves in Figure 6 into frequency domain, with de-convolution of the incident NEMP.

[Abscissa = Frequency (GHz), Ordinate =  $|E_x/E_{inc}|$ ]

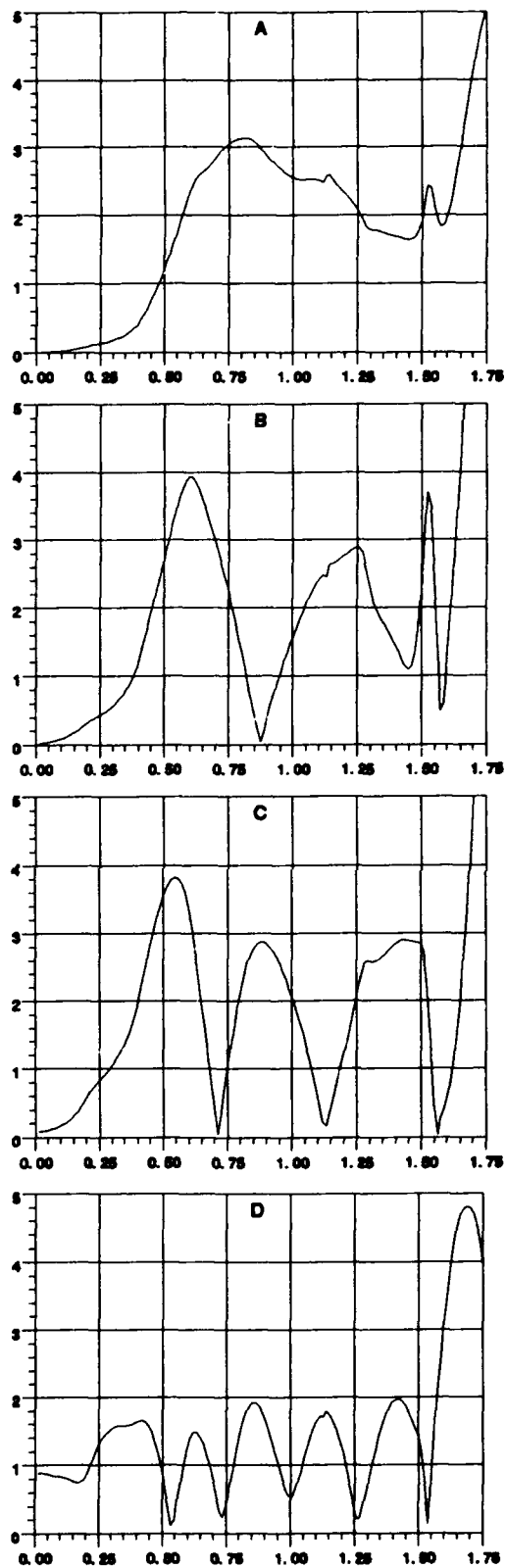


Figure 10. Fourier transforms of the curve in Figure 7 into frequency domain, with de-convolution of the incident Gaussian pulse.

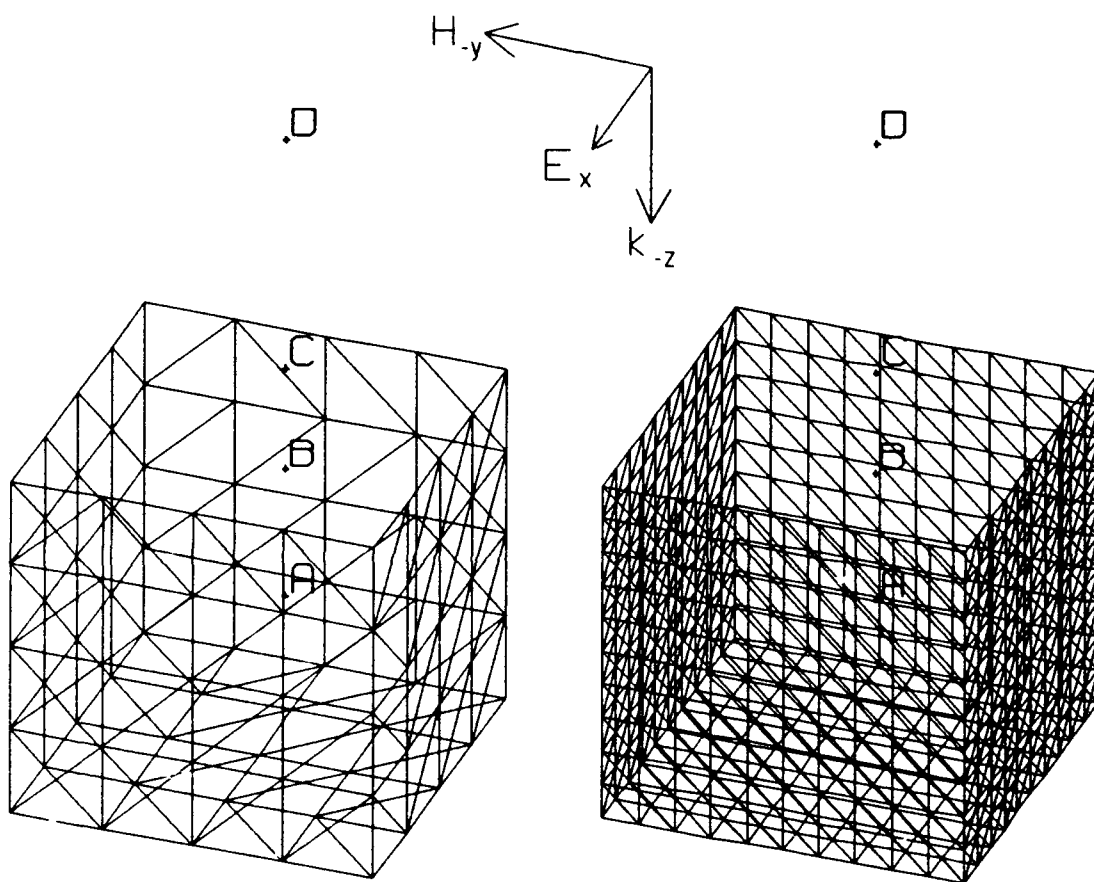


Figure 11. Two resolutions of the open-topped patch-model box used in EFIE frequency-domain studies

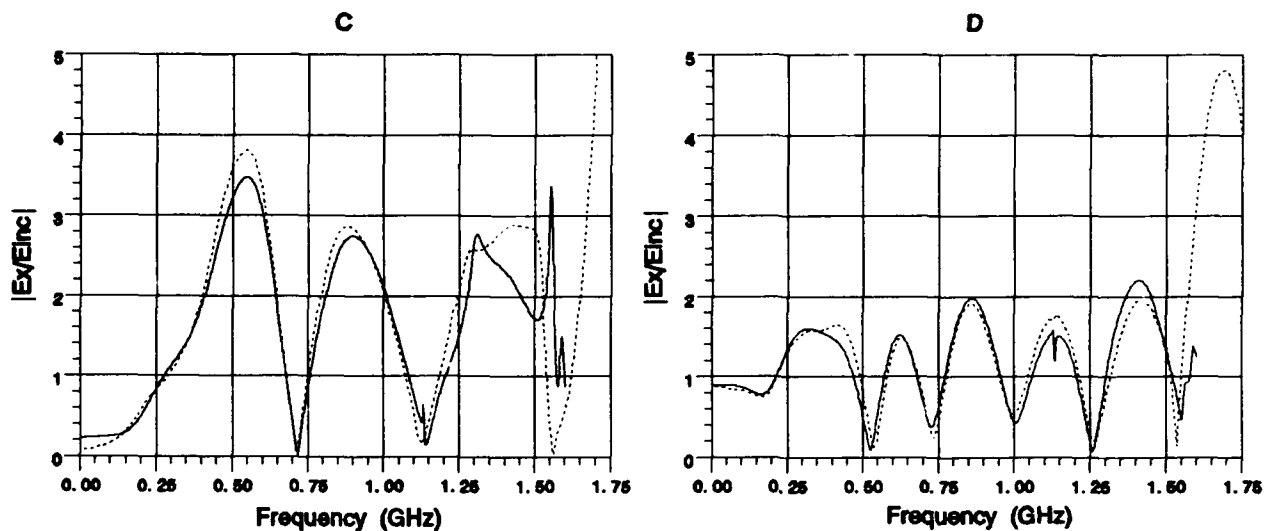


Figure 12. Comparison in frequency domain: Ex-field versus frequency at the field points C and D. Solid curve = EFIE, dashed curve = FFT(FDTD)

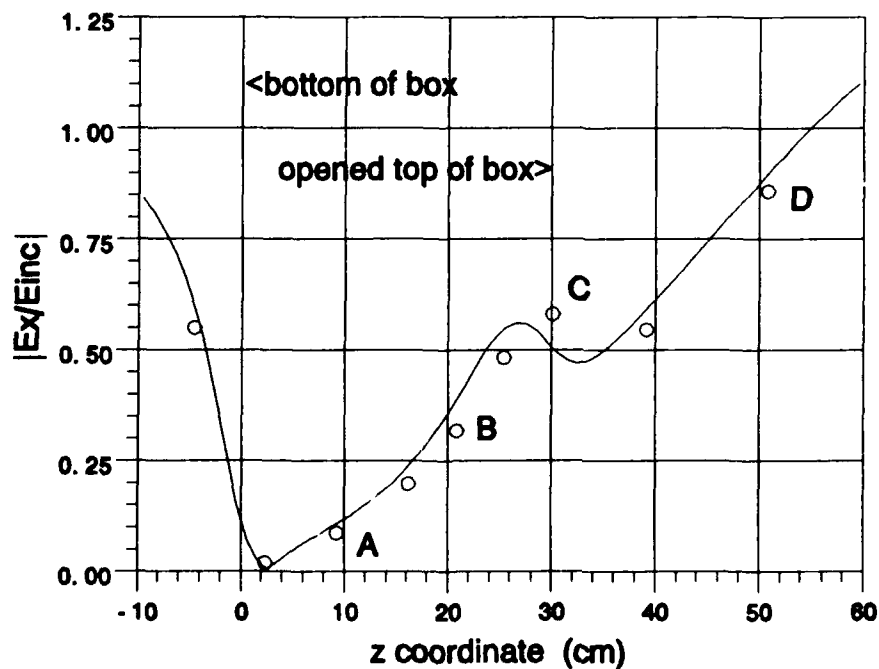


Figure 13. Comparison in frequency domain: Ex-field versus location at the frequency 200 MHz. Solid curve = EFIE, circles = FFT(FDTD)

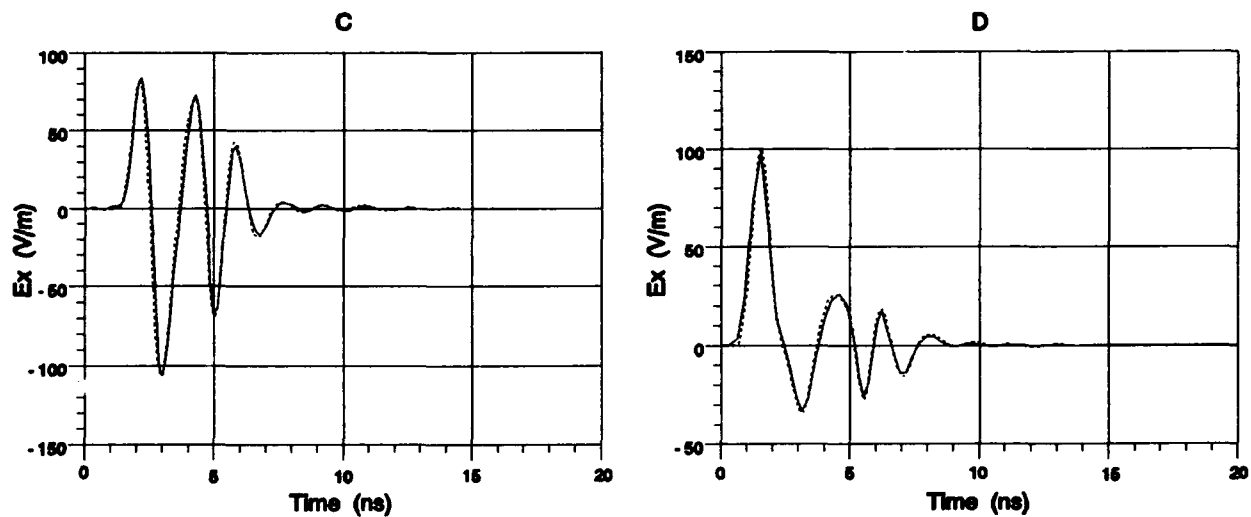


Figure 14. Comparison in time domain: Ex-field at the field points C and D.  
Solid curve = IFFT(EFIE)\*Gaussian, dashed curve = FDTD Gaussian

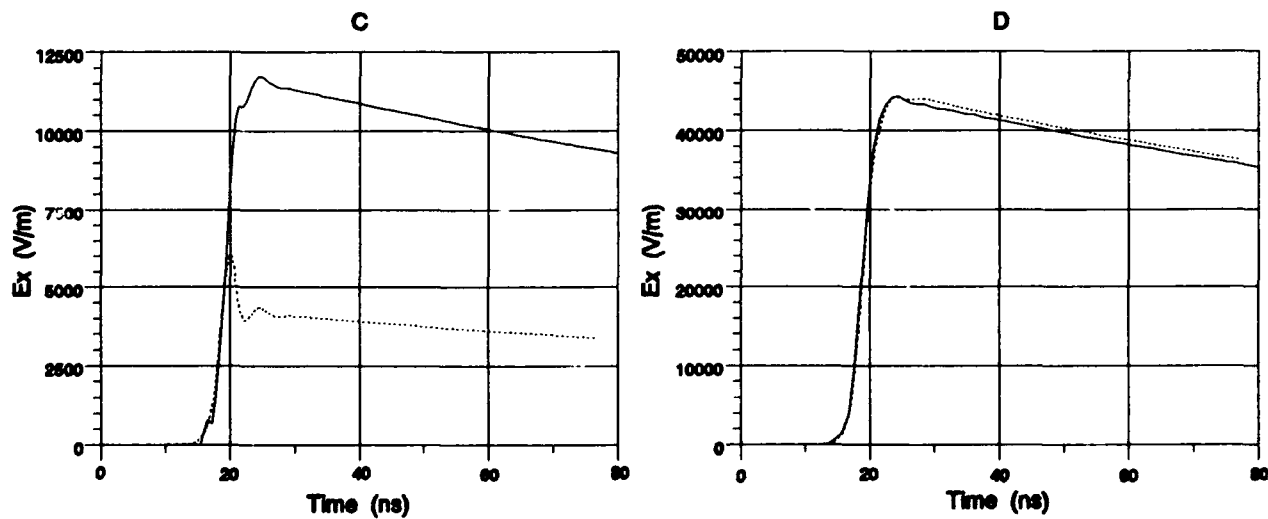


Figure 15. Comparison in time domain: Ex-field at the field points C and D.  
Solid curve = IFFT(EFIE)\*NEMP, dashed curve = FDTD NEMP

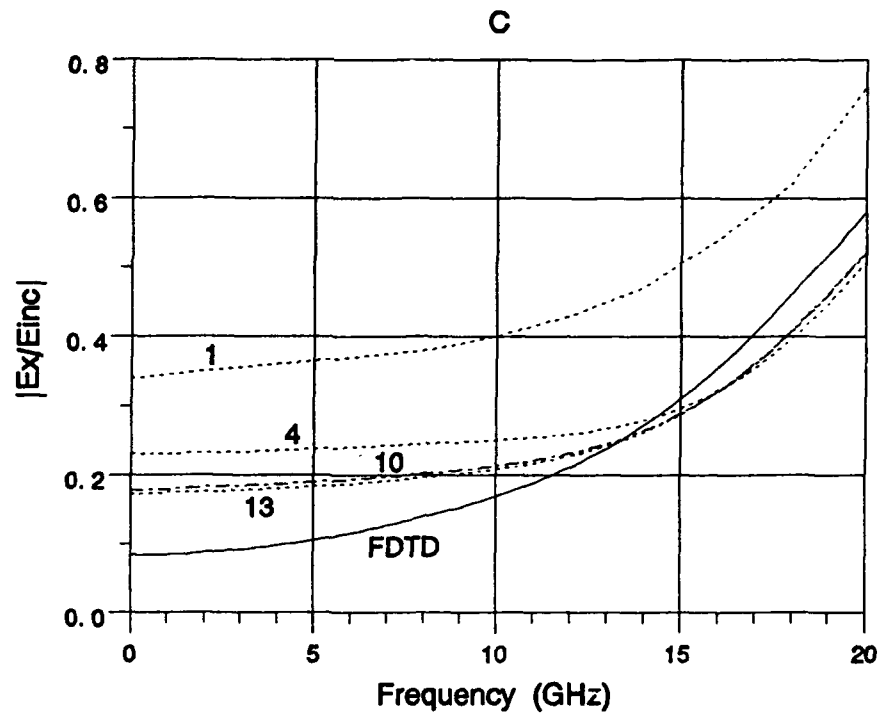


Figure 16. FFT(FDTD) compared to EFIE, in frequency domain at the field point C, with various resolutions of the patch-model box. The numbers on the dashed curves refer to the number of equal divisions of the 0.3m cubic edge.

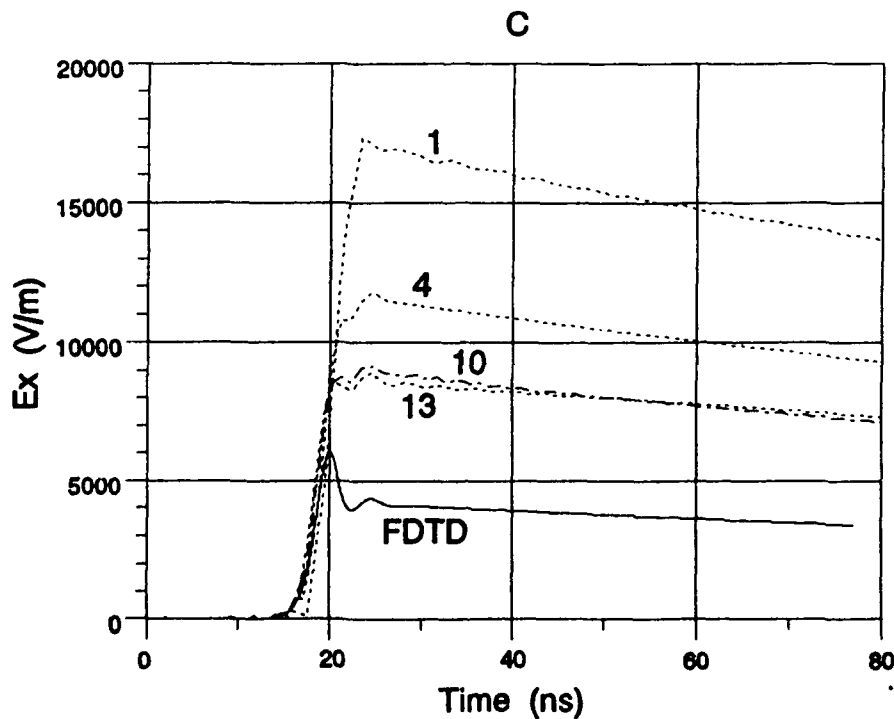


Figure 17. Inverse Fourier transforms of the curves in Figure 16: FDTD compared to IFFT(EFIE)\*NEMP in time domain at the field point C. (The numbers on the dashed curves refer to the number of equal divisions of the 0.3m cubic edge.)

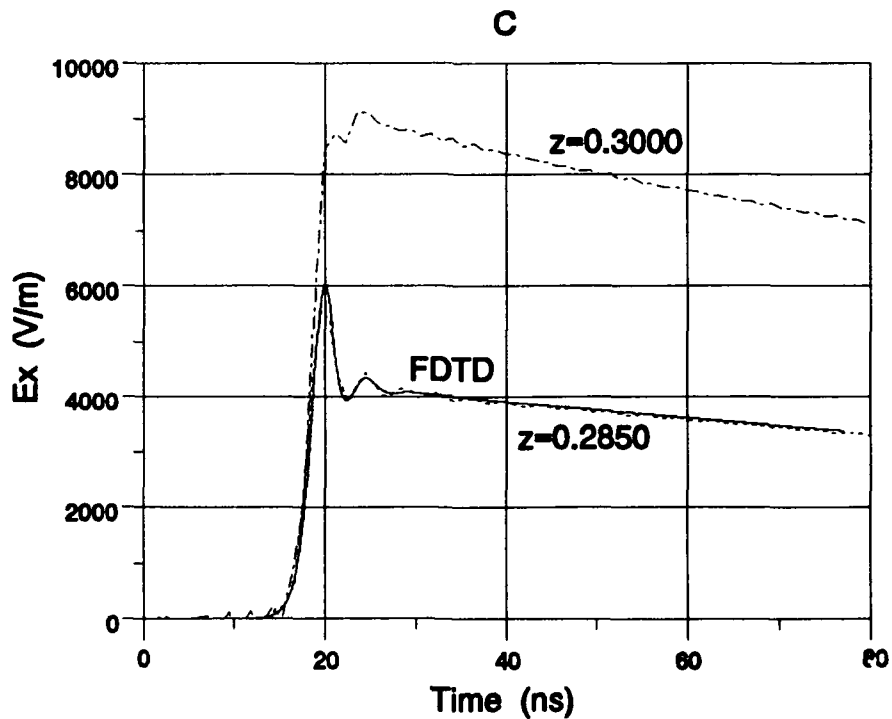
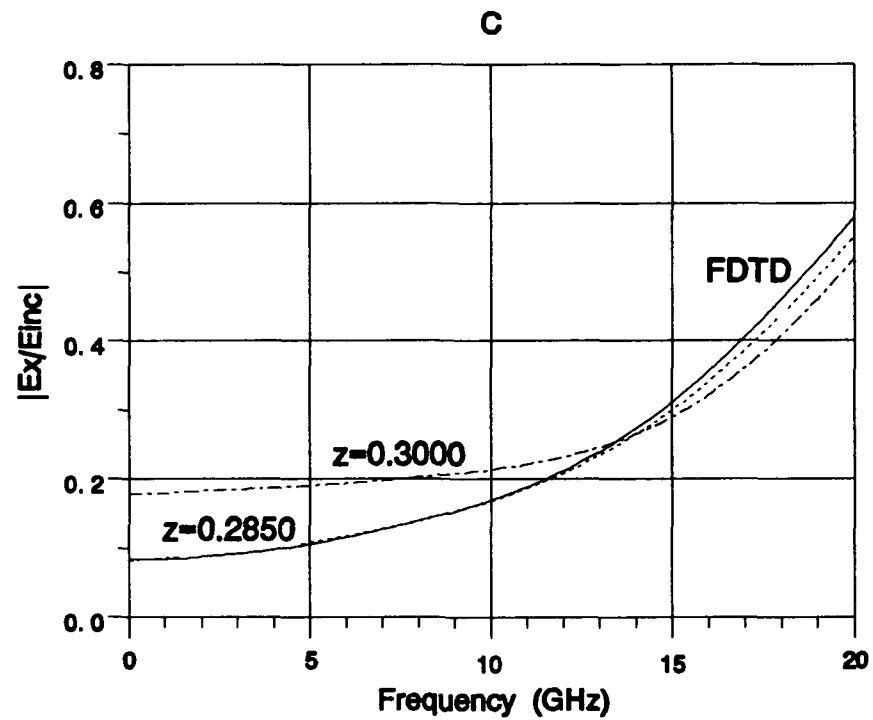


Figure 18. FDTD compared to EFIE in frequency and time domains. The best match for the FDTD field point C = (0.0577,0.1385,0.3000) is the EFIE field point C' = (0.0577,0.1385,0.2850).



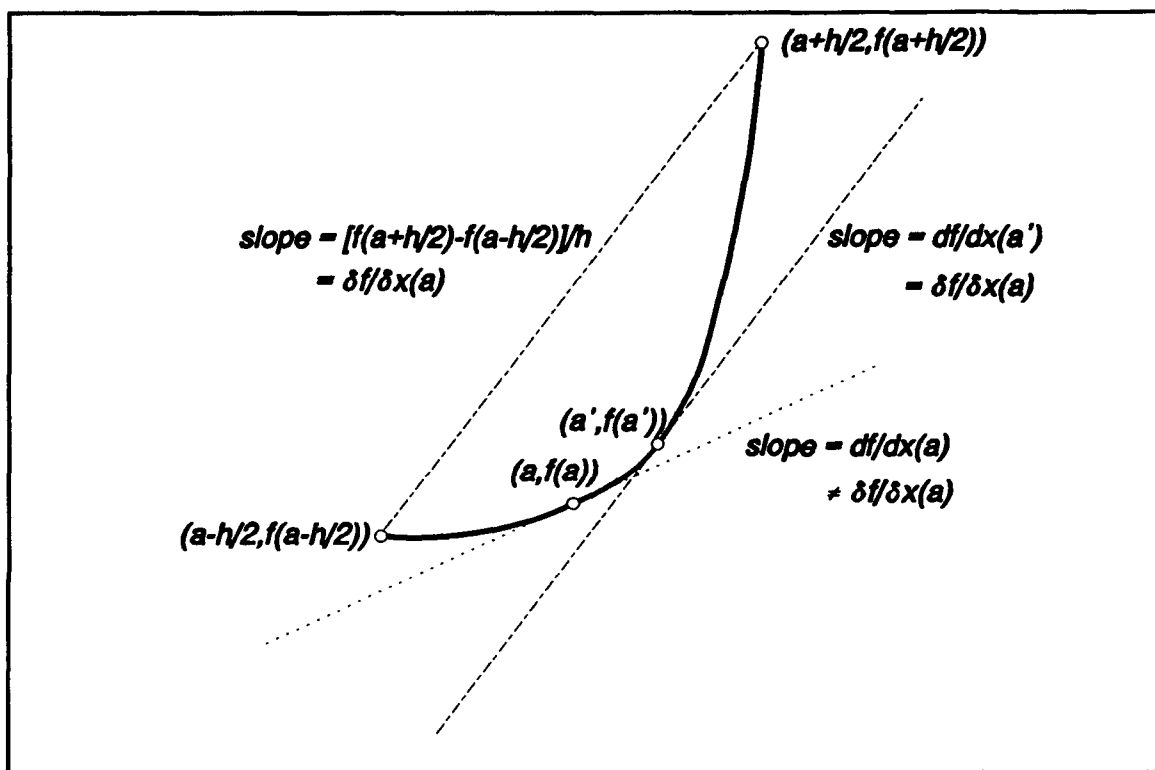


Figure 19. Illustration of the mean-value theorem for derivatives and "discretization error"

# **Validation of the Numerical Electromagnetics Code (NEC) for Antenna Wire Elements in Proximity to Earth**

**M. M. Weiner  
The MITRE Corporation  
Bedford, MA 01730-1420**

## **ABSTRACT**

This paper summarizes recent MITRE efforts to validate the NEC-3 and NEC-GS versions of the Numerical Electromagnetics Code (NEC) developed by Lawrence Livermore National Laboratory for predicting the performance of antenna wire elements in close proximity to flat earth. In an early version (NEC-1), the effect of the air-ground interface was included by applying a plane-wave Fresnel reflection coefficient approximation to the field of a point source. The NEC-2 version, while still retaining the Fresnel reflection coefficient model as an option, provides a more accurate ground model by numerically evaluating Sommerfeld integrals. The version NEC-3 extends the NEC-2 version to cases for bare wire segments below the air-earth interface. Version NEC-GS utilizes rotational symmetry to provide a more efficient version of NEC-3 for the case of a monopole element with a uniform radial wire ground-screen (GS).

Results of the various versions are compared with each other and with other models. The input-output format of the NEC-GS version is discussed. It is concluded that the NEC-3 Sommerfeld integral option in the NEC-GS version is the best available model for monopole elements with electrically small radial-wire ground planes.

## **SECTION 1**

### **INTRODUCTION**

The Numerical Electromagnetics Code (NEC) is a method-of-moments computer program developed by Lawrence Livermore National Laboratory (LLNL) for predicting the performance of wire-element antennas above or buried in flat earth [1, 2]. In an early version (NEC-1), the effect of the air-ground interface was included by applying a plane-wave Fresnel reflection coefficient approximation to the field of a point source [3, 4]. The NEC-2 version, while still retaining the Fresnel reflection coefficient model as an option, provides a more accurate ground model by numerically evaluating Sommerfeld integrals [1, 2]. Version NEC-3 extends the NEC-2 version to cases where bare wire segments are below the air-earth interface [5]. Version NEC-GS is a more efficient version of NEC-3 for wire antennas that have rotational symmetry in the azimuthal direction, such as a monopole element with a uniform radial-wire groundscreen [6, 7]. Version NEC-3I extends NEC-3 to include the case of insulated wires [8, 9]. The NEC-2 program is available to the public, whereas the NEC-3, NEC-GS, and NEC-3I programs are presently available only to U. S. Department of Defense contractors after completion and approval of a NEC order form obtainable from LLNL.

Code documentation has been produced by LLNL for the NEC-2 version and, in a more limited form, for the NEC-3, NEC-GS, and NEC-3I versions. The NEC-2 documentation consists of the theory and code in volume 1 of reference 1 and a user's guide in volume 2 of reference 1. The NEC-3, NEC-GS, and NEC-3I documentations are in the form of user's

guide supplements given in references 5, 7, and 8, respectively. The NEC-2 user's guide and NEC-3 and NEC-GS user's guide supplements give examples of input and output files for most of the options available. Sample input and output files for the NEC-3I program are given in reference 9.

Code validation efforts by LLNL, for antennas near ground, are summarized in reference 10. In addition to several internal consistency checks, reference 10 compares NEC results with those from theoretical models, numerical codes, and to a lesser extent with actual measurements.

The present paper reports recent MITRE efforts to validate the NEC-3 and NEC-GS programs. Results of the various versions are compared with each other and with other models. The input-output format of the NEC-GS version is discussed. Validation results by MITRE are described in sections 2, 3, and 4 for NEC-3 in the Fresnel reflection coefficient option, NEC-3 in the Sommerfeld integral option, and NEC-GS, respectively.

## SECTION 2

### VERSION NEC-3, FRESNEL REFLECTION COEFFICIENT OPTION

#### 2.1 SELECTION OF SQUARE ROOT BRANCH

The NEC-2 and NEC-3 codes, in the Fresnel reflection coefficient option, select the principal value branch of each square root occurring in the equations for the Fresnel reflection coefficients. The question arises as to whether the principal value is the correct branch of the square root, particularly in cases where the effective complex permittivity of the ground plane at the air-ground plane interface has a negative real part. Such cases can occur for wire grids, in free space or in proximity to earth, because the ground plane permeability is conventionally set equal to that of free space.

The Fresnel reflection coefficients  $R_V$  and  $R_H$ , for vertical and horizontal polarizations, respectively, are given by equations (179) and (180) in Volume 1 of reference 1 as

$$R_V = \frac{\cos \theta - Z_R (1 - Z_R^2 \sin^2 \theta)^{1/2}}{\cos \theta + Z_R (1 - Z_R^2 \sin^2 \theta)^{1/2}} \quad (2-1)$$

$$R_H = \frac{(1 - Z_R^2 \sin^2 \theta)^{1/2} - Z_R \cos \theta}{(1 - Z_R^2 \sin^2 \theta)^{1/2} + Z_R \cos \theta} \quad (2-2)$$

Our investigation concludes that the principal value is the correct branch of the quantities  $Z_R \equiv [(\epsilon_1 / \epsilon_o) - j(\sigma_1 / \omega \epsilon_o)]^{-1/2}$  and  $(1 - Z_R^2 \sin^2 \theta)^{1/2} / Z_R$  in equations (2-1) and (2-2) regardless of whether the effective dielectric constant  $\epsilon_1 / \epsilon_o$  is negative, assuming a passive ground medium  $[(\sigma_1 / \omega \epsilon_o) \geq 0]$ . This conclusion follows from the requirements that the

complex wave number  $k = \omega(\epsilon_o \mu_o)^{1/2}/Z_R$  has an argument in the fourth quadrant of the complex plane for a plane wave propagating with a time dependence of the form

$$E = E_o \exp \left[ j \left( \omega t - \vec{k} \cdot \vec{r} \right) \right]$$

and that the magnitude of the Fresnel reflection coefficient does not exceed unity for a plane wave incident from a lossless medium onto a passive medium [13].

In equations (2-1) and (2-2) the principal values of  $(1 - Z_R^2 \sin^2 \theta)^{1/2}$  each satisfy the condition  $|R_V| \leq 1$  for the case of a plane wave incident from a lossless medium onto a passive medium, regardless of whether  $\text{Re}(1/Z_R^2) = \epsilon_1/\epsilon_o$  is positive or negative. Equations (2-1) and (2-2) are in the same form as that given by Stratton [14].

If one divides the numerator and denominator of equation (2-1) by  $Z_R^2$ , one obtains the form given by Reed and Russell [15], namely,

$$R_V = \frac{(1/Z_R)^2 \cos \theta - \left[ (1/Z_R)^2 - \sin^2 \theta \right]^{1/2}}{(1/Z_R)^2 \cos \theta + \left[ (1/Z_R)^2 - \sin^2 \theta \right]^{1/2}} \quad (2-3)$$

In equation (2-3), the principal value of  $\left[ (1/Z_R)^2 - \sin^2 \theta \right]^{1/2}$  satisfies the condition  $|R_V| \leq 1$  for the case of a plane wave incident from a lossless medium onto a passive medium only if

$\text{Re}(1/Z_R^2) = \epsilon_1/\epsilon_o \geq 0$ . For  $\epsilon_1/\epsilon_o < 0$ , the condition  $|R_V| \leq 1$  is satisfied by the

nonprincipal value of  $\left[ (1/Z_R)^2 - \sin^2 \theta \right]^{1/2}$ .

The form of the reflection coefficient given by equation (2-1), unlike that given by equation (2-3), gives correct results for all cases of a wave incident from a lossless medium onto a passive medium if the square roots are restricted to their principal values. The validity of the principal values for the square roots in equations (2-1) and (2-2), subject to this condition, has been confirmed by Burke [16]. In reference 16, please note that a wire grid in

free space has a relative permittivity ( $=$  principal value  $\left[ (Y_g + Y_o)/Y_o \right]^2$ ) whose imaginary part has a conductivity greater than zero and whose real part (the dielectric constant) is negative. The wire grid admittance is given by  $Y_g \approx -jY_o / [(s/\lambda) \ln(s/\pi d)]$  and the free space admittance is given by  $Y_o = (\epsilon_o/\mu_o)^{1/2}$  where  $s$  is the grid spacing ( $s/\lambda \ll 1$ ) and  $d$  is

the wire diameter ( $d/s \ll 1$ ). However, even for the case of a negative dielectric constant and positive conductivity, the principal values of the square roots yield valid results. For that case,  $|R_V|$  equals unity and  $\arg R_V$  differs by 180 degrees from that for a positive dielectric constant and positive conductivity. This result is analogous to the case of a perfect ground plane for which  $|R_V|$  equals unity and  $\arg R_V$  differs by 180 degrees from that for an imperfect ground plane at an angle of incidence equal to 90 degrees.

## 2.2 COMPARISON WITH OTHER MODELS

Fresnel reflection coefficient models for antennas in proximity to earth are generally grossly inaccurate in determining input impedance, radiation efficiency, and power gains unless the ground plane and monopole element current distributions are predetermined by other methods such as the method of moments. However, Fresnel reflection coefficient models are accurate when determining the absolute directivity or directivity pattern for the case of an antenna element in proximity to earth, or a ground plane of infinite extent (see discussion at end of this section). These remarks are applicable not only to the Fresnel reflection option of the NEC-2 and NEC-3 programs, but also to any model which attempts to approximate the ground plane current, originating from a spherical wave source, by that determined from a plane-wave, Fresnel reflection coefficient model.

The antenna element current distribution, in the reflection coefficient option of NEC-3, is determined by considering the mutual impedance between the source antenna element and its ground plane image. The ground plane image is determined by considering the Fresnel reflection coefficient only for the ground plane (or earth) directly below the antenna element. Consequently, ground screens of small density or extent will yield the same reflection coefficient as a ground screen of large density or extent. Furthermore, the Fresnel reflection coefficient model neglects groundscreen edge diffraction and underestimates earth losses, both of which can be significant for small ground planes. For these reasons, the input impedance of an antenna element in proximity to earth is poorly estimated by the reflection coefficient option unless the element has a ground plane of sufficiently large density and extent.

Fresnel reflection coefficient models are grossly inaccurate in computing the radiation efficiency of antennas in close proximity to earth because such models only consider ground losses caused by plane-wave reflection and refraction and ignore spherical-wave generation of a leaky evanescent surface wave that is generated in the air medium in proximity to the air-earth interface. The surface wave, with an evanescent field in the air-medium only, leaks energy into the earth medium but not into the air medium [17, 18]. A comparison of the radiation efficiency  $\eta$  (= ratio of far-field radiated power in air to the input power delivered to the antenna) calculated by the Sommerfeld option of NEC-3 (which considers surface wave ground losses) with that calculated by a Fresnel reflection coefficient model is shown in Table 1 at a frequency of 15 MHz for a vertically polarized thin dipole whose base has zero current and is zero height above CCIR-527-1 classifications of Earth [11]. For medium dry earth, the Sommerfeld option yields numeric radiation efficiencies of 0.104 and 0.304 for element lengths of 0.02 and 0.25 wavelengths, respectively, whereas the Fresnel reflection coefficient model predicts radiation efficiencies of 0.013 and 0.283, respectively. In this example, the Fresnel reflection coefficient model underestimates the radiation efficiency by

**Table 1. Radiation Efficiency of a Vertically Polarized Thin Dipole of Length  $h$  Whose Base has Zero Current and is Zero Height above Earth,  $f = 15$  MHz**

CCIR-527-1 Earth Classification ( $\epsilon_r$ $\sigma$ S/m)	Radiation Efficiency $\eta$ (numeric)							
	$h/\lambda = 0.01$	$h/\lambda = 0.02$	$h/\lambda = 0.05$	$h/\lambda = 0.10$	$h/\lambda = 0.15$	$h/\lambda = 0.20$	$h/\lambda = 0.25$	$h/\lambda = 0.50$
(1) Perfect Ground (1, $\infty$ )								
Sommerfeld	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Fresnel	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
(2) Sea Water (70, 5)								
Sommerfeld*	2.33 ( $10^{-3}$ )	0.019	0.220	0.599	0.724	0.758	0.766	0.708
Fresnel	8.67 ( $10^{-2}$ )	0.451	0.703	0.763	0.778	0.780	0.775	0.707
(3) Fresh Water (80, $3.0 \times 10^{-2}$ )								
Sommerfeld*	7.34 ( $10^{-4}$ )	5.94 ( $10^{-3}$ )	0.080	0.308	0.432	0.474	0.483	0.410
Fresnel	2.82 ( $10^{-3}$ )	2.05 ( $10^{-2}$ )	0.145	0.302	0.379	0.417	0.434	0.396
(4) Wet Ground (30, $1.0 \times 10^{-2}$ )								
Sommerfeld*	4.42 ( $10^{-4}$ )	3.57 ( $10^{-3}$ )	0.050	0.217	0.328	0.370	0.381	0.321
Fresnel	9.46 ( $10^{-4}$ )	7.38 ( $10^{-3}$ )	0.073	0.194	0.263	0.301	0.321	0.305
(5) Medium Dry Ground (15, $1.0 \times 10^{-3}$ )								
Sommerfeld*	1.30 ( $10^{-3}$ )	0.104	0.116	0.296	0.347	0.357	0.355	0.304
Fresnel	1.73 ( $10^{-3}$ )	1.30 ( $10^{-2}$ )	9.05 ( $10^{-2}$ )	0.175	0.222	0.252	0.272	0.283
(6) Very Dry Ground (3, $1.0 \times 10^{-4}$ )								
Sommerfeld*	1.05 ( $10^{-3}$ )	8.25 ( $10^{-3}$ )	8.89 ( $10^{-2}$ )	0.214	0.250	0.260	0.265	0.283
Fresnel	6.30 ( $10^{-4}$ )	4.94 ( $10^{-3}$ )	5.08 ( $10^{-2}$ )	0.126	0.163	0.186	0.204	0.264
(7) Pure Water, 20 °C (80, $1.7 \times 10^{-2}$ )								
Sommerfeld*	3.63 ( $10^{-2}$ )	0.202	0.500	0.544	0.542	0.534	0.523	0.437
Fresnel	6.65 ( $10^{-2}$ )	0.126	0.215	0.324	0.392	0.429	0.448	0.420
(8) Ice, -1 °C (3, $9.0 \times 10^{-5}$ )								
Sommerfeld*	1.16 ( $10^{-3}$ )	9.14 ( $10^{-3}$ )	9.53 ( $10^{-2}$ )	0.218	0.252	0.262	0.266	0.283
Fresnel	6.99 ( $10^{-3}$ )	5.47 ( $10^{-3}$ )	5.43 ( $10^{-2}$ )	0.129	0.164	0.187	0.205	0.264
(9) Ice, -10 °C (3, $2.7 \times 10^{-5}$ )								
Sommerfeld*	3.84 ( $10^{-3}$ )	0.0283	0.175	0.254	0.267	0.270	0.271	0.286
Fresnel	2.30 ( $10^{-4}$ )	0.0167	9.60 ( $10^{-2}$ )	0.150	0.175	0.194	0.209	0.267
(10) Average Land (10, $5.0 \times 10^{-3}$ )								
Sommerfeld*	1.16 ( $10^{-4}$ )	1.34 ( $10^{-3}$ )	1.95 ( $10^{-2}$ )	0.106	0.193	0.241	0.260	0.236
Fresnel	1.97 ( $10^{-4}$ )	1.59 ( $10^{-3}$ )	2.10 ( $10^{-2}$ )	8.96 ( $10^{-2}$ )	0.148	0.185	0.207	0.219
(11) Free Space (1, 0)								
Sommerfeld	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Fresnel	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

\* More accurate result

Sommerfeld = NEC-3 program in Sommerfeld option N = 9 segments,  $b/\lambda = 10^{-5}$ , voltage excitation at 5th segment

Fresnel = NEC-3 program in Fresnel reflection coefficient option, N = 9 segments,  $b/\lambda = 10^{-5}$ , voltage excitation at 5th segment

88 percent and 7 percent for element lengths of 0.02 and 0.25 wavelengths, respectively. For seawater, the Sommerfeld option yields numeric radiation efficiencies of 0.019 and 0.708 for element lengths of 0.02 and 0.25 wavelengths, respectively, whereas the Fresnel reflection coefficient model predicts radiation efficiencies of 0.451 and 0.707, respectively. In that example, the Fresnel reflection coefficient model overestimates the radiation efficiency by 2273 percent and 0.14 percent, respectively. The Fresnel reflection coefficient model is therefore inappropriate for computing radiation efficiency for antennas in close proximity to earth. The error in using the Sommerfeld integral option is no more than 23 per cent for the worst case, as discussed in section 3.2.

Despite the inadequacy of Fresnel reflection coefficient models for estimating the input impedance and radiation efficiency of antenna elements in close proximity to earth, such models are accurate in estimating the antenna's absolute directivity and directivity pattern for the case of an antenna element in proximity to earth (or to a ground plane of infinite extent) and for an antenna element whose ground plane current distribution is pre-determined by other methods. The directivity, computed by the NEC-3 Sommerfeld option, Richmond's method of moments [19,21], and a Fresnel reflection coefficient model, are compared in Table 2 for the case of a vertically polarized quarter-wave monopole element on medium dry earth. Each model gives the same directivity to within 0.04 dB at a given angle of incidence.

However, even for a case where directivity is correctly given by a Fresnel reflection coefficient model, the power gain (= directivity x radiation efficiency) may be incorrect because the radiation efficiency may be grossly inaccurate.

## SECTION 3

### VERSION NEC-3, SOMMERFELD INTEGRAL OPTION

#### 3.1 LLNL VALIDATION EFFORTS

LLNL has compared numerical results for the input impedance and electric field of a sloping base long-wire antenna over conducting Earth, obtained from NEC-3 in the Sommerfeld integral option, with measurements by Breakall and Christman [10]. Predicted versus measured values differed approximately by 25 to 100 percent for input resistance,  $\pm 30$  ohms about 0 ohms for input reactance, and 1 to 9 dB  $\mu\text{V/m}$  for the electric field.

#### 3.2 MODIFIED RADIATION EFFICIENCY OF A VERTICALLY POLARIZED, HERTZIAN DIPOLE IN PROXIMITY TO DIELECTRIC EARTH

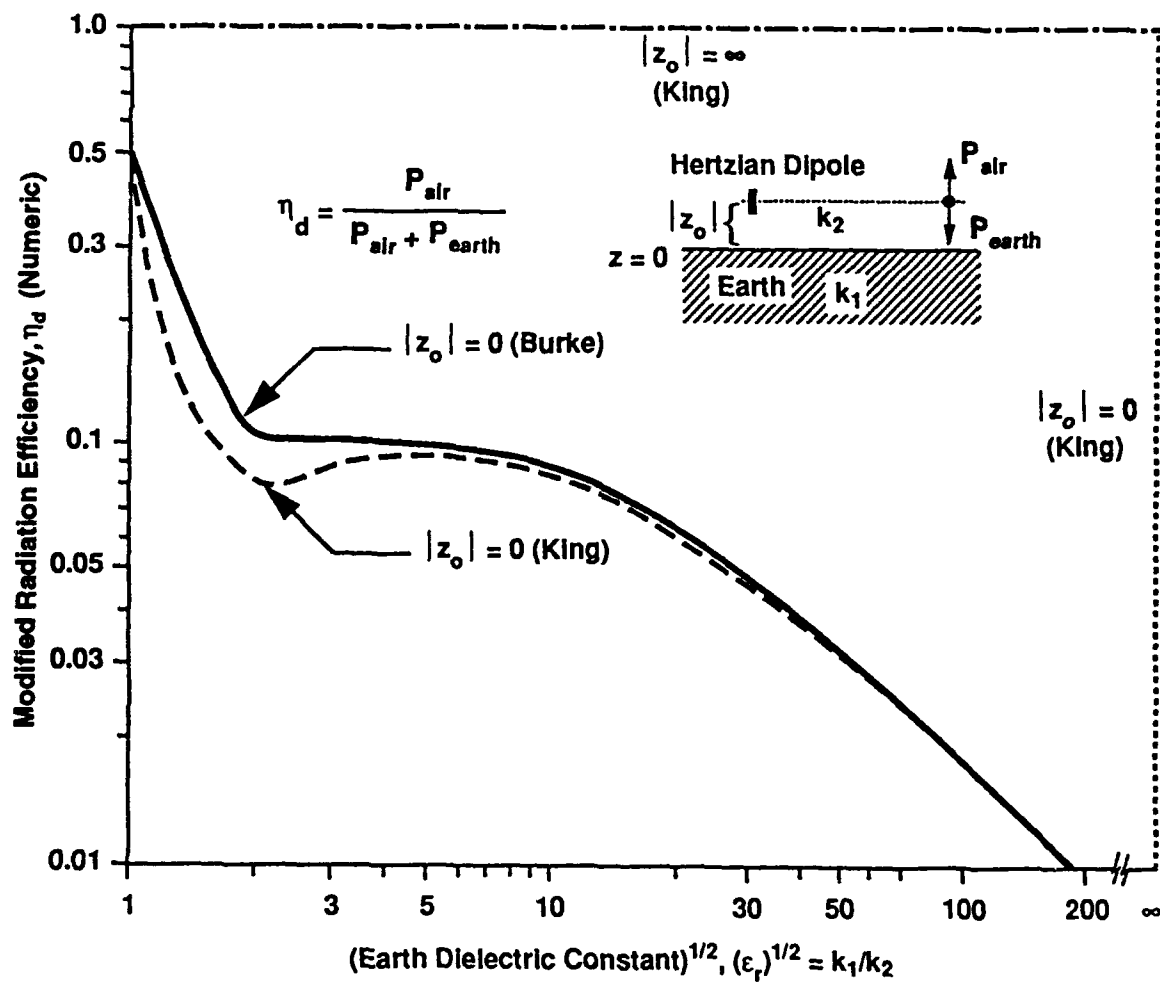
NEC-3 results by Burke [20], for the modified radiation efficiency  $\eta_d$  (defined in Figure 1) of an electrically short, vertical dipole above dielectric Earth, are compared in Figures 1 and 2 with King's analytical results for a Hertzian vertical dipole [18] obtained by integrating the vertical component of the Poynting vector along a far-field line parallel to the air-Earth interface. The two models give similar results for sufficiently large values of the Earth dielectric constant, but differ by 15 percent for the Earth dielectric constant  $\epsilon_r = 9$  (or  $|k_1/k_2| = 3$ ) when the dipole is at zero height above the Earth. The results of King are approximate because his analytical model is subject to the condition  $\epsilon_r \gg 1$  (or equivalently

Table 2. Directivity of a Thin, Vertically Polarized Quarter-wave Monopole Element Resting on Medium Dry Earth, 15 MHz

Angle of Incidence, $\theta$ (deg)	Directivity (dBi)			Angle of Incidence, $\theta$ (deg)	Directivity (dBi)		Modified Images (Fresnel Coefficient) <sup>3</sup>
	NEC-3 (Sommerfeld Option) <sup>1</sup>	Richmond (Method of Moments) <sup>2</sup>	Modified Images (Fresnel Coefficient) <sup>3</sup>		NEC-3 (Sommerfeld Option) <sup>1</sup>	Richmond (Method of Moments) <sup>2</sup>	
0	-∞	-∞	-∞	46	3.80	3.78	3.80
2	-22.90	-22.88	-22.88	48	4.07	4.05	4.07
4	-16.88	-16.86	-16.86	50	4.31	4.29	4.31
6	-13.35	-13.34	-13.34	52	4.53	4.50	4.53
8	-10.86	-10.85	-10.84	54	4.71	4.69	4.71
10	-8.92	-8.91	-8.91	56	4.86	4.84	4.86
12	-7.34	-7.33	-7.33	58	4.98	4.96	4.98
14	-6.00	-5.99	-5.99	60	5.07	5.04	5.06
16	-4.85	-4.84	-4.84	62	5.11	5.08	5.11
18	-3.83	-3.82	-3.82	64	5.12	5.09	5.11
20	-2.92	-2.92	-2.91	66	5.07	5.04	5.07
22	-2.10	-2.10	-2.09	68	4.98	4.95	4.98
24	-1.36	-1.36	-1.35	70	4.82	4.79	4.82
26	-0.68	-0.68	-0.67	72	4.59	4.56	4.59
28	-0.06	-0.06	-0.05	74	4.27	4.23	4.26
30	0.52	0.52	0.53	76	3.83	3.80	3.83
32	1.05	1.05	1.06	78	3.24	3.21	3.24
34	1.54	1.54	1.55	80	2.44	2.40	2.44
36	2.00	1.99	2.00	82	1.33	1.29	1.32
38	2.42	2.41	2.42	84	-0.29	-0.33	-0.29
40	2.81	2.80	2.81	86	-2.86	-2.90	-2.87
42	3.17	3.16	3.17	88	-7.85	-7.89	-7.85
44	3.50	3.48	3.50	90	-∞	-∞	-∞

1. Voltage excitation source between the earth and the base of the element.
2. Program RICHMD4 with small ground plane of normalized radius  $2\pi a/\lambda = 0.025$  wave numbers.
3. Program MODIFIED IMAGES assumes Fresnel reflection coefficient and sinusoidal current distribution on element.
4. Medium Dry Earth ( $\epsilon_r = 15$ ,  $\sigma = 1.0 \times 10^{-3}$ ).





**Figure 1. Modified Radiation Efficiency of a Vertical Hertzian Dipole at Zero Height Above Dielectric Earth**

Hertzian dipole at height  $|z_0|$  in air ( $k_2$ )  
over a dielectric half-space ( $k_1 = k_2 \sqrt{\epsilon_r}$ )

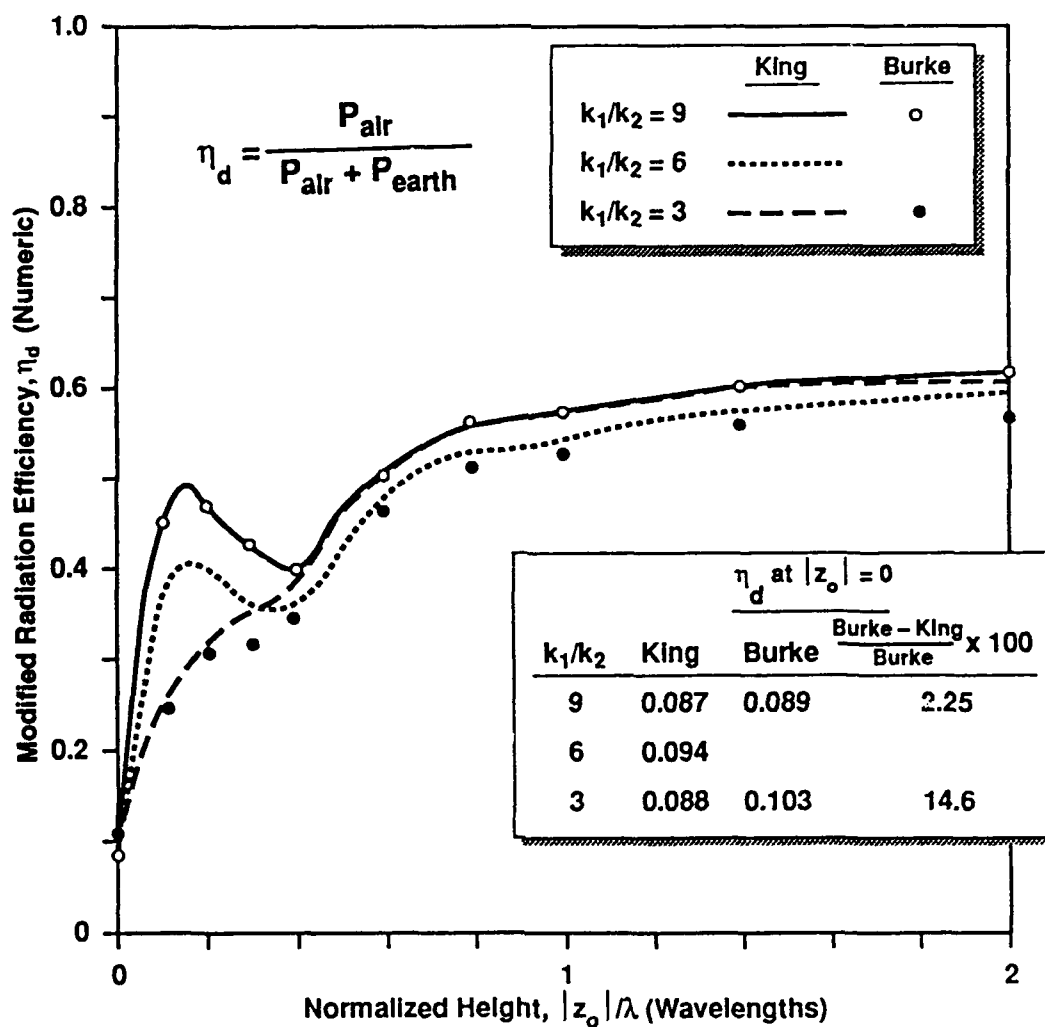


Figure 2. Modified Radiation Efficiency of a Vertical Hertzian Dipole at Various Heights Above Dielectric Earth

$k_1/k_2 \geq 3$ ) except for the condition  $\epsilon_r = 1$  which is treated separately. The NEC-3 results are for a vertical dipole of half-length =  $10^{-4}$  wavelengths and radius =  $10^{-6}$  wavelengths at a height  $|z_o|$  above earth measured from the center-feed of the dipole. In this paper the convention is followed that lower-case  $z$  designates the Earth's vertical position with respect to the antenna ground plane (or base of the antenna element, in the absence of a ground plane) and upper-case  $Z$  designates impedance.

The case of a lossless antenna element over dielectric Earth provides an excellent opportunity for testing the accuracy of the antenna input current,  $I$ , computed by the Sommerfeld integral option of the NEC-3 code. The antenna power gain  $\bar{G}$  averaged over the radiation sphere (solid angle of  $4\pi$  steradians) is defined as

$$\bar{G} = P_{rad}/P_{in} = (P_{air} + P_{earth})/P_{in} \quad (3-1)$$

where

- $P_{rad}$  = total far-field radiated power =  $P_{air} + P_{earth}$
- $P_{air}$  = far-field radiated power in the air
- $P_{earth}$  = far-field radiated power in dielectric Earth
- $P_{in}$  = input power delivered to the antenna =  $(1/2) Re(VI^*)$
- $V$  = input voltage complex amplitude (set equal to 1 volt) in the NEC program for a steady-state sinusoidal source.
- $I^*$  = conjugate input current complex amplitude (amperes) that is solved for in the NEC-3 program. The asterisk denotes "conjugate."

The quantities  $P_{air}$  and  $P_{earth}$ , for an antenna element with azimuthal symmetry, is given by

$$P_{air} = \left[ r^2 / (2Z_o) \right] 2\pi \int_0^{\pi/2} (\vec{E} \cdot \vec{E}^*) \sin \theta \, d\theta \quad (3-2)$$

$$P_{earth} = \left[ r^2 / (2Z_o) \right] 2\pi \int_{\pi}^{\pi/2} (\vec{E} \cdot \vec{E}^*) \sin \theta \, d\theta \quad (3-3)$$

where

- $r$  = distance from the antenna element to the far-field point  $P(r, \theta, \phi)(m)$
- $E$  = electric field intensity at the far-field point  $P(r, \theta, \phi)(V/m)$
- $Z_o = (\mu_o / \epsilon_o)^{1/2}$  = free space wave impedance (ohms)

For a lossless antenna over dielectric Earth, the average power gain  $\bar{G}$  equals 1, if there are no errors in the NEC-3 program and the computer has infinite precision. Assuming that the computer has sufficient precision and that the integration steps in equations (3-2) and (3-3)

are sufficiently small, then any deviation of  $\bar{G}$  from unity is a measure of the accuracy of the current  $I$  computed by the NEC-3 program. The reason is that  $P_{in}$  is proportional to  $I$  whereas,  $P_{air}$  and  $P_{earth}$  are proportional to  $|I|^2$  (because  $E$  is proportional to  $I$ ).

The quantity  $\bar{G}$ , as computed by the NEC-3 program, is a weak function of the number  $N$  of segments (or current variables) chosen to represent the antenna element. Whenever one uses the method-of-moments, too coarse a segmentation results in poor accuracy due to undersampling the current distribution. Too fine a segmentation can again result in poor accuracy because of round-off errors caused by the finite precision of the computer. The element segmentation, for vertical dipole and monopole elements above Earth, is shown for a voltage excitation source in Figure 3. For a thin, electrically short dipole at a height

$|z_o|/\lambda = 0.4$  above dielectric Earth  $\bar{G}$  differs from unity by 1.4 percent for  $N=5$  and 0.1 percent for  $N = 101$  (see Table 3). For the same dipole at a height

$|z_o|/\lambda = 0.0001$  above the same dielectric Earth,  $\bar{G}$  differs from unity by 22.3 percent for  $N = 11$  and 22.6 percent for  $N = 101$ . Even though the element segment length for  $N = 101$  in Table 3 is one-half the recommended minimum segment length relative to the segment radius (see section 4.2), the results are not significantly different than for  $N = 51$ .

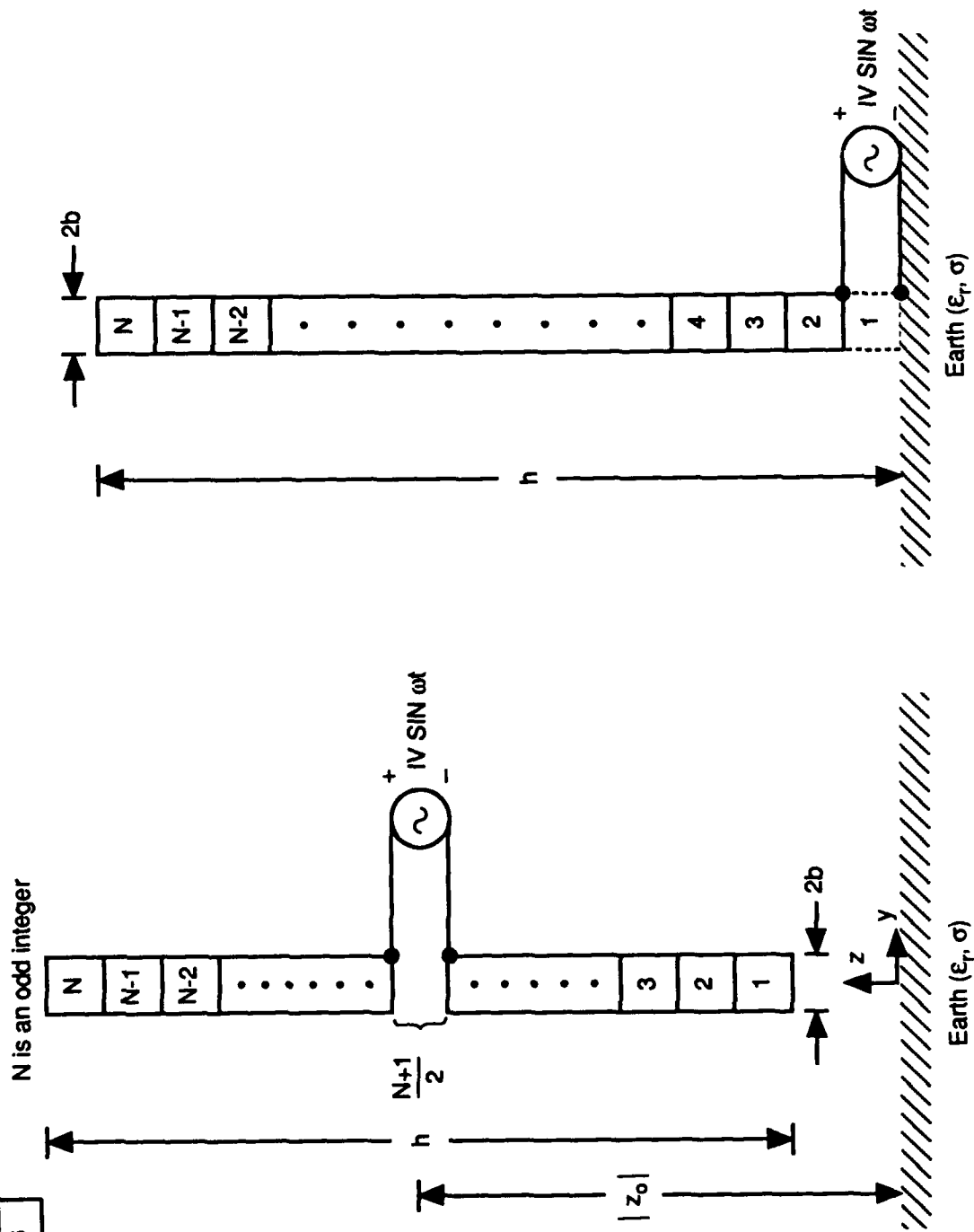
The difference of  $\bar{G}$  from unity increases with increasing Earth dielectric constant and decreasing element height above earth. For a dipole at a height  $|z_o|/\lambda = 0.0001$ ,  $\bar{G}$  differs from unity by 7.6 percent for  $\epsilon = 1$  and 40.2 percent for  $\epsilon = 81$  (see Table 4). For a dipole above dielectric Earth with  $\epsilon = 9$ ,  $\bar{G}$  differs from unity by 22.6 percent for  $|z_o|/\lambda = 0.0001$  and 0.7 percent for  $|z_o|/\lambda = 2.0$  (see Table 5).

The differences of  $\bar{G}$  from unity in Tables 3 through 6 indicate that the NEC-3 program has inaccuracies as much as 25 percent or more in computing input current, input impedance, and input power for an electrically short antenna element in close proximity to Earth. These in-accuracies do not apply to the computation of modified efficiency  $\eta_d$  but would also affect the computation of radiation efficiency if the antenna element were in proximity to lossy Earth since radiation efficiency is a function of the absolute accuracy of the input current.

For dielectric Earth the modified radiation efficiency is not dependent upon the absolute accuracy of the input current since both ( $P_{air}$  and  $P_{earth}$ ) are proportional to the same computed value of input current. Therefore, the modified radiation efficiency computed by NEC-3 for dielectric Earth is accurate to within the precision of the computer and the size of the integration steps  $\Delta\theta$  of the far-field power density. In NEC-3, the modified radiation efficiency  $\eta_d$  is computed for dielectric Earth as the quotient of  $P_{air}$  divided by ( $P_{air} + P_{earth}$ ), namely:

$$\eta_d = \text{modified radiation efficiency} = \eta / \bar{G} = P_{air} / (P_{air} + P_{earth}) \quad (3-4)$$

IY0006



(a) Dipole with Voltage Source Excitation (b) Monopole with Voltage Source Excitation

Figure 3. Segmentation of Vertical Antenna Element Above Earth

**Table 3. Effect of Number of Dipole Segments on Average Power Gain and Radiation Efficiency Computed by Program NEC-3 for an Electrically-Small, Vertical Dipole at Heights  $|z_0/\lambda| = 0.4$  and  $0.0001$  above Dielectric Earth ( $\epsilon_r = 9$ ,  $\sigma = 0$ )**

No. of Segments, N	*Average Power Gain, $\bar{G} = (P_{air} + P_{earth})/P_{in}$		Radiation Efficiency, $\eta = P_{air}/P_{in}$		Modified Radiation Efficiency, $\eta_d = P_{air}/(P_{air} + P_{earth}) = \eta / \bar{G}$	
	$ z_0/\lambda  = 0.4$	$ z_0/\lambda  = 0.0001$	$ z_0/\lambda  = 0.4$	$ z_0/\lambda  = 0.0001$	$ z_0/\lambda  = 0.4$	$ z_0/\lambda  = 0.0001$
5	0.9858	—	0.3356	—	0.3404	—
11	0.9963	1.223	0.3392	0.1260	0.3404	0.1030
21	0.9983	—	0.3399	—	0.3404	—
31	0.9989	1.226	0.3400	0.1263	0.3404	0.1030
41	0.9989	—	0.3400	—	0.3404	—
51	0.9989	—	0.3401	—	0.3404	—
81	0.9990	—	0.3401	—	0.3404	—
101	0.9990	1.226	0.3401	0.1263	0.3404	0.1030

Dipole length  $h/\lambda = 2 \times 10^{-4}$

Dipole radius  $b/\lambda = 1 \times 10^{-6}$

Integration step  $\Delta\theta = 1.0$  deg,  $0 \leq \theta \leq 90$  deg;  $0.1$  deg,  $90 \leq \theta \leq 180$  deg

$P_{air}$ ,  $P_{earth}$  = far-field radiated powers in air and dielectric earth, respectively

$P_{in} = (1/2) \text{Re}(VI^*) = (1/2) V \text{Re } I^*$

\*  $\bar{G} = 1$ , for a loss-less element over dielectric earth, if there were no errors in NEC-3 program and the computer had infinite precision.

**Table 4. Effect of Earth Dielectric Constant on Average Power Gain and Radiation Efficiency Computed by Program NEC-3 for an Electrically-Small, Vertical Dipole at Height  $|z_0/\lambda| = 0.0001$  above Dielectric Earth ( $\sigma = 0$ )**

Dielectric Constant, $\epsilon_r$	*Average Power Gain, $\bar{G} = (P_{air} + P_{earth})/P_{in}$	Radiation Efficiency, $\eta = P_{air}/P_{in}$	Modified Radiation Efficiency, $\eta_d = P_{air}/(P_{air} + P_{earth}) = \eta / \bar{G}$
1.0	0.9997	0.4998	0.5000
2.25	1.0763	0.1493	0.1387
4.0	1.1301	0.1272	0.1126
9.0	1.2257	0.1263	0.1030
16.0	1.2826	0.1300	0.1014
25.0	1.3096	0.1311	0.1001
36.0	1.3442	0.1308	0.0973
49.0	1.3565	0.1294	0.0954
64.0	1.3848	0.1272	0.0919
81.0	1.4017	0.1244	0.0888
100.0	1.4026	0.1214	0.0866
400.0	1.4840	0.0919	0.0620
900.0	1.5258	0.0724	0.0475
1600.0	1.5295	0.0593	0.0388
2500.0	1.5321	0.0501	0.0327
3600.0	1.5308	0.0433	0.0283
4900.0	1.5278	0.0382	0.0250
6400.0	1.5223	0.0342	0.0225
8100.0	1.5148	0.0309	0.0204

Dipole length  $h/\lambda = 2 \times 10^{-4}$ , dipole radius  $b/\lambda = 1 \times 10^{-6}$ , no. of dipole segments  $N = 31$

Integration step  $\Delta\theta = 1.0$  deg,  $0 \leq \theta \leq 90$  deg;  $0.1$  deg,  $90 < \theta \leq 180$  deg.

\*  $\bar{G} = 1$ , for a loss-less element over dielectric earth, if there were no errors in NEC-3 program and the computer had infinite precision

**Table 5. Effect of Earth Dipole Height on Average Power Gain and Radiation Efficiency Computed by Program NEC-3 for an Electrically-Small, Vertical Dipole above Dielectric Earth ( $\epsilon_r = 9$ ,  $\sigma = 0$ )**

Height Above Earth, $ z_0 /\lambda$	*Average Power Gain, $\bar{G}$ $= (P_{air} + P_{earth})/P_{in}$	Radiation Efficiency $\eta = P_{air}/P_{in}$	Modified Radiation Efficiency, $\eta_d$ $= P_{air}/(P_{air} + P_{earth})$ $= \eta / \bar{G}$
0.0001	1.2257	0.1263	0.1030
0.0003	1.2257	0.1266	0.1033
0.001	1.2165	0.1269	0.1043
0.003	1.1895	0.1276	0.1073
0.01	1.1053	0.1301	0.1177
0.03	1.0096	0.1498	0.1483
0.1	0.9988	0.2410	0.2413
0.2	1.0007	0.2971	0.2969
0.3	0.9987	0.3078	0.3082
0.4	0.9987	0.3400	0.3404
0.6	0.9970	0.4609	0.4623
0.8	0.9971	0.5082	0.5097
1.0	0.9958	0.5207	0.5229
1.4	0.9945	0.5463	0.5493
2.0	0.9928	0.5593	0.5633

Number of dipole segments  $N = 31$

Dipole length  $h/\lambda = 2 \times 10^{-4}$

Dipole radius  $b/\lambda = 1 \times 10^{-6}$

Integration step  $\Delta\theta = 1.0$  deg,  $0 \leq \theta \leq 90$  deg; 0.1 deg,  $90 < \theta < 180$  deg

\*  $\bar{G} = 1$ , for a loss-less element over dielectric earth, if there were no errors in NEC-3 program and the computer had infinite precision

**Table 6. Effect of Number of Dipole Segments on Average Power Gain and Radiation Efficiency Computed by Program NEC-3 for an Electrically-Small, Vertical Monopole Whose Base Rests on Dielectric Earth ( $\epsilon_r = 9$ ,  $\sigma = 0$ )**

No. of Segments, $N$	*Average Power Gain, $\bar{G}$ $= (P_{air} + P_{earth})/P_{in}$	Radiation Efficiency, $\eta = P_{air}/P_{in}$	Modified Radiation Efficiency, $\eta_d = P_{air}/(P_{air} + P_{earth})$ $= \eta / \bar{G}$
5	1.1703	0.1205	0.1030
11	1.1871	0.1223	0.1030
21	1.1877	0.1223	0.1030
31	1.1837	0.1219	0.1030

Monopole length  $h/\lambda = 2 \times 10^{-4}$

Monopole radius  $b/\lambda = 1 \times 10^{-6}$

Integration step  $\Delta\theta = 1.0$  deg,  $0 \leq \theta \leq 90$  deg; 0.1 deg,  $90 < \theta \leq 180$  deg.

\*  $G = 1$ , for a loss-less element over dielectric earth if there were no error in NEC-3 program and the computer had infinite precision.

where

$$\eta = \text{radiation efficiency} = P_{air}/P_{in}$$

$$\bar{G} = (P_{air} + P_{earth})/P_{in} \text{ [see equation (3-1)]}$$

The modified radiation efficiency  $\eta_d$  computed by NEC-3 is shown in Tables 3 through 5 for a dipole above dielectric earth. In Table 3, the modified radiation efficiency is independent of the number of element segments as is also the case for an electrically-small vertical monopole element whose base rests on earth (see Table 6). The modified radiation efficiencies of an electrically-small vertical dipole and monopole, of the same length and radius and whose bases rest on earth, should be identical. This result is achieved by the

NEC-3 program (compare Table 3 for  $|z_o|/\lambda = 0.0001$  with Table 6). If the monopole element in Table 6 is increased to a quarter-wave length with 25 segments, the average power gain  $\bar{G} = 0.9990$  [22].

The Figure 3(b) geometry for a monopole element driven by a voltage excitation source between its base and Earth should be used with caution in the NEC-3 program when the Earth is lossy. Although convergent results of modified radiation efficiency as a function of the number of element segments were obtained for dielectric Earth, nonconvergent results were obtained for lossy Earth. However, in the computation of directivity, such a model gives valid results for lossy Earth (see Table 2)

### **3.3 PROPAGATION CONSTANT OF CURRENT ON BARE, HORIZONTAL WIRE (BEVERAGE ANTENNA) ABOVE LOSSY EARTH**

Recent measurements of the propagation constant of the current on a Beverage antenna comprising a bare, horizontal wire 12 inches above medium dry earth and terminated in a load impedance have been reported at a frequency of 18 MHz [23]. The measurements are in excellent agreement with an analytical model of King [24] and in poorer agreement with numerical results from the NEC-3 program. Burke has recently reported that NEC-2 (and NEC-3) predictions of the propagation constant are in good agreement with a theoretical model by Olsen, Kuester, and Chang [30].

### **3.4 INPUT IMPEDANCE, DIRECTIVITY PATTERN, AND ABSOLUTE GAIN OF A MONOPOLE ELEMENT WITH A BURIED RADIAL-WIRE GROUND PLANE**

Measurements by Harnish, Lee, and Hagn of the input impedance of a monopole element with a buried radial-wire ground plane are in reasonable agreement with NEC-3 predictions [31].



## SECTION 4

### VERSION NEC-GS

#### 4.1 APPLICABILITY OF ANTENNA GEOMETRY

Version NEC-GS is a more efficient version of NEC-3 for wire antennas that have rotational symmetry in the azimuthal direction. Examples of antennas with such a geometry are a vertical, electrically thick, dipole element; a vertical, electrically thick, monopole element; and a monopole element whose ground plane consists of  $N$  uniformly-spaced radial wires, all of which may be in proximity to earth.

NEC-GS is a more efficient version for such a geometry because the input parameter specification is simplified and the matrix size (total number of wire segments or current variables) is reduced. For example, instead of specifying the coordinates for each segment of  $N$  radial wires, it is only necessary to specify the segment coordinate for a single wire. Furthermore, the matrix size for  $N$  radial wires with  $k$  segments/wire is reduced from  $kN$  to  $k$  when the number of rotations  $M$  equals  $N$ . The reduced matrix size enables NEC-GS to model antennas with larger wires and a greater number of wires than can be modeled by NEC-3.

#### 4.2 INPUT PARAMETER SPECIFICATION

Input parameter guidelines are given in reference 7. The following guidelines [26] may also be of interest to the user.

Wire intersections are assumed to be connected if two wires are within each other by an amount of  $1/1000$  of a segment length.

Horizontal wires on the air side of the earth interface should not approach the earth's surface to within the greater of  $10^{-6} \lambda$  or 2 to 3 times the wire radius.

A monopole segment that is connected to a horizontal wire should be at least as short as the height of the horizontal wire above the earth's surface.

The physical junction of several radial wires with a vertical element is modeled as a singular point (a node) without regard as to whether the radial wires are conically tapered so that they are physically able to fit around the vertical element.

The wire currents at a node are constrained to satisfy Kirchhoff's current law without regard for current leakage into the earth.

The format for the field of the input parameters, as illustrated on page 5 of reference 7, should be meticulously followed. For example, in the GR card that specifies the integer number of ground radials, the omission of the concluding comma increases the number of radials by a factor of ten.

In the NEC-3 and NEC-GS programs, the segment length should be at least four times longer than the segment radius. If not, the extended kernel option (IK card) should be used for segment lengths as small as one segment radius.

The difference in radii of two adjoining wire segments (or two wires at a junction) should be minimized. A method for minimizing the difference in radii is the tapering of segment radii along one of the adjoining wires.

A rotational model may be used to represent a vertical element of radius  $b$  by a cage of  $M$  vertical elements each of radius  $b_w$  along a circumference of radius  $b$ . Best results are obtained by  $b_w = b/M$  so that the vertical elements have the same total surface area as the original element [27, 28]. Rotational model representations of a vertical dipole element and a monopole element with a radial wire ground plane — all in proximity to earth — are shown in Figures 4 and 5, respectively. In Figure 5, the number of rotations  $M$  is equal to the number of radial wires, and the radius of the rotational vertical elements is equal to the radius  $b_w$  of the radial wires.

### 4.3 INTERPRETATION OF OUTPUT PARAMETERS

When the rotational model is not used ( $M = 1$ ), the output parameters represent those of the physical antenna. However, when the rotational model is used, the output parameters are those of the rotational elements and not those of the physical antenna. The algebraic operations required on the rotational model output parameters to obtain the output parameters for the physical antenna are summarized in Table 7.

### 4.4 COMPARISON WITH OTHER MODELS

#### 4.4.1 LLNL Validation Efforts

LLNL has compared NEC-GS numerical results with theoretical results based on the compensation theorem by J. R. Wait and W. A. Pope for the input impedance of a quarter-wave monopole on a buried, radial-wire ground plane [10]. Good agreement was obtained between the two models only for those cases where implementation of the compensation theorem is expected to be valid, namely, for groundscreens of sufficient density (the number  $N$  of radial wires is large) and of sufficient extent (the length  $a$  of the radial wires are at least a wavelength in Earth). Unlike the NEC-GS method-of-moments model, the present implementation of the compensation theorem never solves the current on the ground plane, but instead assumes that the current distribution is the same as for a perfect ground plane. The inadequacy of the present implementation of the compensation theorem to yield accurate results of input impedance for small ground planes in proximity to earth was also pointed out by J. H. Richmond [19] when comparing his method-of-moments results for disk ground planes with results based on the compensation theorem by Wait and Surtees [29].

#### 4.4.2 Comparison with NEC-3

This subsection compares numerical results obtained from NEC-GS with those obtained from NEC-3.

The test case, in the NEC-GS user's guide [7], is for a monopole element with six buried radial ground wires that have the same radius as that of the monopole element. Test case numerical results, obtained from NEC-GS with no rotations ( $M = 1$ ), agree to within 0.01 percent of those obtained from NEC-3 (see Table 8).

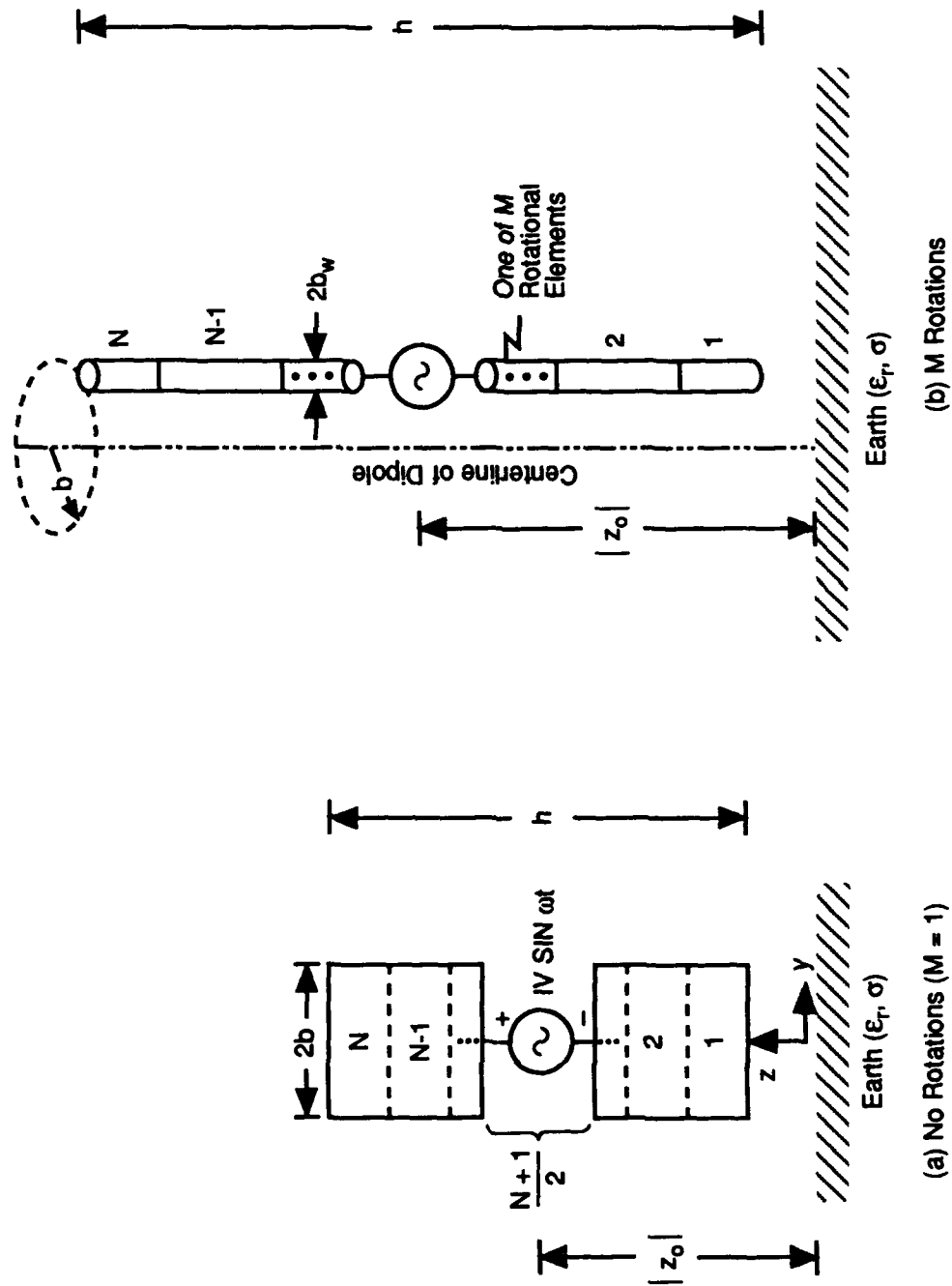
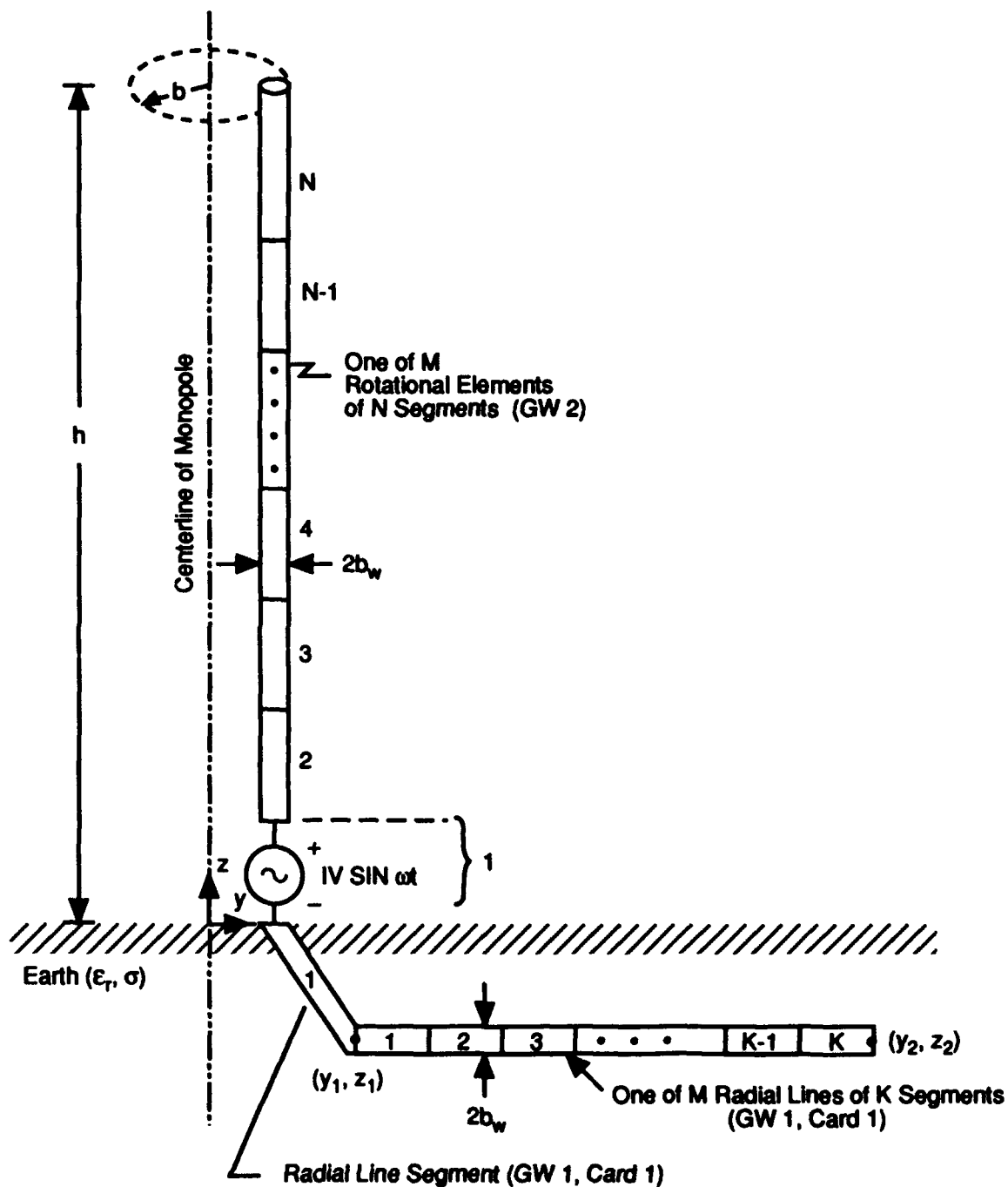


Figure 4. Rotational Model Representation of a Vertical Dipole Element in Proximity to Earth



**Figure 5. Rotational Model Representation of a Monopole Element of Radius  $b$  with a Ground Plane of  $M$  Radial Wires of Radius  $b_w$  in Proximity to Earth**

**Table 7. Algebraic Operations to Obtain Output Parameters of Physical Antenna When Using a NEC-GS Rotational Model with M Rotations**

Output Parameter of Physical Antenna	Operation Required on the Rotational Model Output Parameter
Current on vertical element (amperes)	Multiply by M
Current on radial wire (amperes)	As printed out
Input impedance of vertical element (ohms)	Divide by M
Input admittance of vertical element (mhos)	Multiply by M
Radiation efficiency* (numeric)	Divide by M
Gain (dB)	Subtract $10 \log_{10} M$

\* Radiation efficiency = one-half of printed out value of average power gain for cases when the antenna is in proximity to lossy earth

**Table 8. Comparison of Numerical Results Obtained from NEC-GS ( $M = 1$ ) with Those Obtained from NEC-3 for Monopole Element with a Buried Radial-Wire Ground Plane**

Test case in G. J. Burke, "User's Guide Supplement for NEC-GS," Lawrence Livermore National Laboratory, Report UCRL-MA-107572, June, 1991.

Output parameter	Numerical value	
	NEC-GS	NEC-3
Element input current (amperes)	1.4279 E-2 - j7.7917 E-3	1.4278 E-2 - j7.7928 E-3
Radial wire input current (amperes)	2.279 E-3 - j1.375 E-3	2.2824 E-3 - j1.3754 E-3
Element input impedance (ohms)	5.3964 E+1 + j2.9446 E+1	5.39628 E+1 + j2.94524 E+1
Radiation efficiency (numeric)	0.291	0.291
Peak power gain	-0.27 dB	-0.27 dB
Direction of peak power gain	65 deg	65 deg

NEC-GS numerical results are compared with NEC-3 results in Tables 9 through 11 for a dipole element and for thin and thick monopole elements with six buried radial ground wires, respectively. The corresponding NEC-GS rotational model geometries for these antennas are given in Figures 4 and 5, respectively. The numerical results for NEC-GS with no rotations ( $M = 1$ ) are almost identical in all cases to NEC-3 numerical results as is to be expected, since the model geometries are identical. Numerical results for NEC-GS rotational models ( $M \geq 2$ ) have mixed agreement with NEC-3 numerical results.

The NEC-GS rotational model for the dipole element yields numerical results that are in almost exact agreement with those from NEC-3 when  $M = 100$ , corresponding to the case when the total surface area of the dipole rotational elements is equal to that of the physical dipole (see Table 9). For  $M = 4$ , the element input current differs by 25 percent from that from NEC-3 although other output parameters are in close agreement with NEC-3.

**Table 9. Comparison of Dipole Element Numerical Results Obtained from NEC-GS Rotational Models ( $M = 1, 4, 8, 12, 16, 100$ ) with Those Obtained from NEC-3.**

$$h/\lambda = 0.250, b/\lambda = 1.667 \times 10^{-4}, b_w/\lambda = 1.667 \times 10^{-6}, |z_o|/l = 0.130, N = 21 \text{ segments} \\ \epsilon_r = 10.0, \sigma = 0.01 \text{ S/m}, f = 5 \text{ MHz } (\lambda = 60 \text{ m})$$

Model		Element Input Current (Amperes after Multiplying by M)	Element Input Impedance (Ohms after Dividing by M)	Radiation Efficiency (Numeric after Dividing by M)	Power Gain (after Sub- tracting $10 \log_{10} M$ ) Peak (dB) Direction (deg)
NEC-3		0.712 E-4 + j0.155 E-2	0.296 E+2 - j0.644 E+3	0.306	0.12 dB 67 deg
NEC-GS	M = 1	0.715 E-4 + j0.155 E-2	0.296 E+2 - j0.642 E+3	0.306	0.12 dB 67 deg
	M = 4	0.532 E-4 + j0.134 E-2	0.294 E+2 - j0.742 E+3	0.307	0.13 dB 68 deg
	M = 8	0.632 E-4 + j0.146 E-2	0.294 E+2 - j0.682 E+3	0.306	0.12 dB 67 deg
	M = 12	0.665 E-4 + j0.150 E-2	0.295 E+2 - j0.665 E+3	0.306	0.12 dB 67 deg
	M = 16	0.681 E-4 + j0.152 E-2	0.295 E+2 - j0.657 E+3	0.306	0.12 dB 67 deg
	M = 100	0.712 E-4 + j0.155 E-2	0.295 E+2 - j0.643 E+3	0.306	0.12 dB 67 deg

The NEC-GS rotational model with  $M = 6$  for a thin monopole element with six buried radial ground wires yields numerical results that differ from those for NEC-3 by eighteen percent for the monopole input current, by nineteen percent for the radial-wire input current, by twelve percent for the input impedance, by eleven percent for the radiation efficiency, and by 0.5 dB for the peak power gain (see Table 10). The total surface area of the monopole rotational elements is six percent ( $Mb_w/b = 6 \times 1.667 \times 10^{-6} / 1.667 \times 10^{-4} = 0.06$ ) of that of the physical monopole.

The NEC-GS rotational model with  $M = 6$ , for a thick monopole element and radial wires whose diameters are ten times larger than those in Table 10, yields numerical results that differ from NEC-3 by six percent for the monopole input current, by seven percent for the radial-wire input current, by two percent for the input impedance, by six percent for the radiation efficiency, and 0.3 dB for the peak power gain (see Table 11). It is not clear why close agreement with NEC-3 (within six percent) is obtained for the thick monopole element whose rotational elements have the same total surface area relative to the physical element as for the thinner element in Table 10.

**Table 10. Comparison of Numerical Results Obtained from NEC-GS Rotational Models ( $M = 1, 6$ ) of a Thin Monopole Element with Radial-Wire Ground Plane with Those Obtained from NEC-3.**

$h/\lambda = 0.250$ ,  $b/\lambda = 1.667 \times 10^{-4}$ ,  $N = 10$  segments (GW, 2)  
 $b_w/\lambda = 1.667 \times 10^{-6}$ ,  $N = 14$  segments (GW1, Card 2),  $(y_1, z_1) = (0.8 \text{ m}, -0.05 \text{ m})$   
 $(y_2, z_2) = (12.0 \text{ m}, -0.05 \text{ m})$ ,  $\epsilon_r = 10.0$ ,  $\sigma = 0.01 \text{ S/m}$ ,  $f = 5 \text{ MHz}$  ( $\lambda = 60 \text{ m}$ )

Output Parameter	Numerical Value		
	NEC-3	NEC-GS ( $M = 1$ )	NEC-GS ( $M = 6$ )
Element input current (amperes, after multiplying by $M$ )	$0.129 \text{ E-1} - j0.704 \text{ E-2}$	$0.129 \text{ E-1} - j0.704 \text{ E-2}$	$0.106 \text{ E-1} - j0.680 \text{ E-2}$
Radial wire input current (amperes, as printed out)	$0.195 \text{ E-2} - j0.131 \text{ E-2}$	$0.195 \text{ E-2} - j0.131 \text{ E-2}$	$0.158 \text{ E-2} - j0.124 \text{ E-2}$
Element input impedance (ohms, after dividing by $M$ )	$5.980 \text{ E+1} + j3.272 \text{ E+1}$	$5.980 \text{ E+1} + j3.272 \text{ E+1}$	$6.695 \text{ E+1} + j4.307 \text{ E+1}$
Radiation efficiency (numeric, after dividing by $M$ )	0.263	0.263	0.233
Peak power gain (dB, after subtracting $10 \log_{10} M$ )	-0.72	-0.71	-1.24
Direction of peak gain (degrees)	65	65	65

**Table 11. Comparison of Numerical Results Obtained from NEC-GS Rotational Models (M = 1, 6) of a Thick Monopole Element with Radial-Wire Ground Plane with Those Obtained from NEC-3**

$h/\lambda = 0.250$ ,  $b/\lambda = 1.667 \times 10^{-3}$ ,  $N = 10$  segments (GW2)  
 $b_w/\lambda = 1.667 \times 10^{-5}$ ,  $N = 14$  segments (GW1, card 2),  $(y_1, z_1) = (8.0 \text{ m}, -0.05 \text{ m})$   
 $(y_2, z_2) = (12.0 \text{ m}, -0.05 \text{ m})$ ,  $\epsilon_r = 10.0$ ,  $\sigma = 0.01 \text{ S/m}$ ,  $f = 5 \text{ MHz}$ ,  $(\lambda = 60 \text{ m})$

Output Parameter	Numerical Value		
	NEC-3	NEC-GS (M = 1)	NEC-GS (M = 6)
Element input current (amperes, after multiplying by M)	0.130 E-1 - j0.686 E-2	0.130 E-1 - j0.686 E-2	0.122 E-1 - j0.689 E-2
Radial wire input current (amperes, as printed out)	0.202 E-1 - j0.125 E-2	0.202 E-1 - j0.125 E-2	0.187 E-1 - j0.125 E-2
Element input impedance (ohms, after dividing by M)	6.015 E+1 + j3.171 E+1	6.016 E+1 + j3.171 E+1	6.223 E+1 + j3.523 E+1
Radiation efficiency (numeric, after dividing by M)	0.279	0.279	0.263
Peak power gain (dB, after subtracting 10 log <sub>10</sub> M)	-0.44	-0.44	-0.71
Direction of peak gain (degrees)	65	65	65

#### 4.4.3 Richmond's Method-of-Moments

NEC-GS results [33,34] (for a 128 radial-wire ground plane) are in close agreement with results from Richmond's method-of-moments program RICHMOND4 [19,21,35] (for a disk ground plane) when computing the radiation efficiency of a quarter-wave monopole element with a small ground plane on or in close proximity to medium dry Earth (see Figure 6).



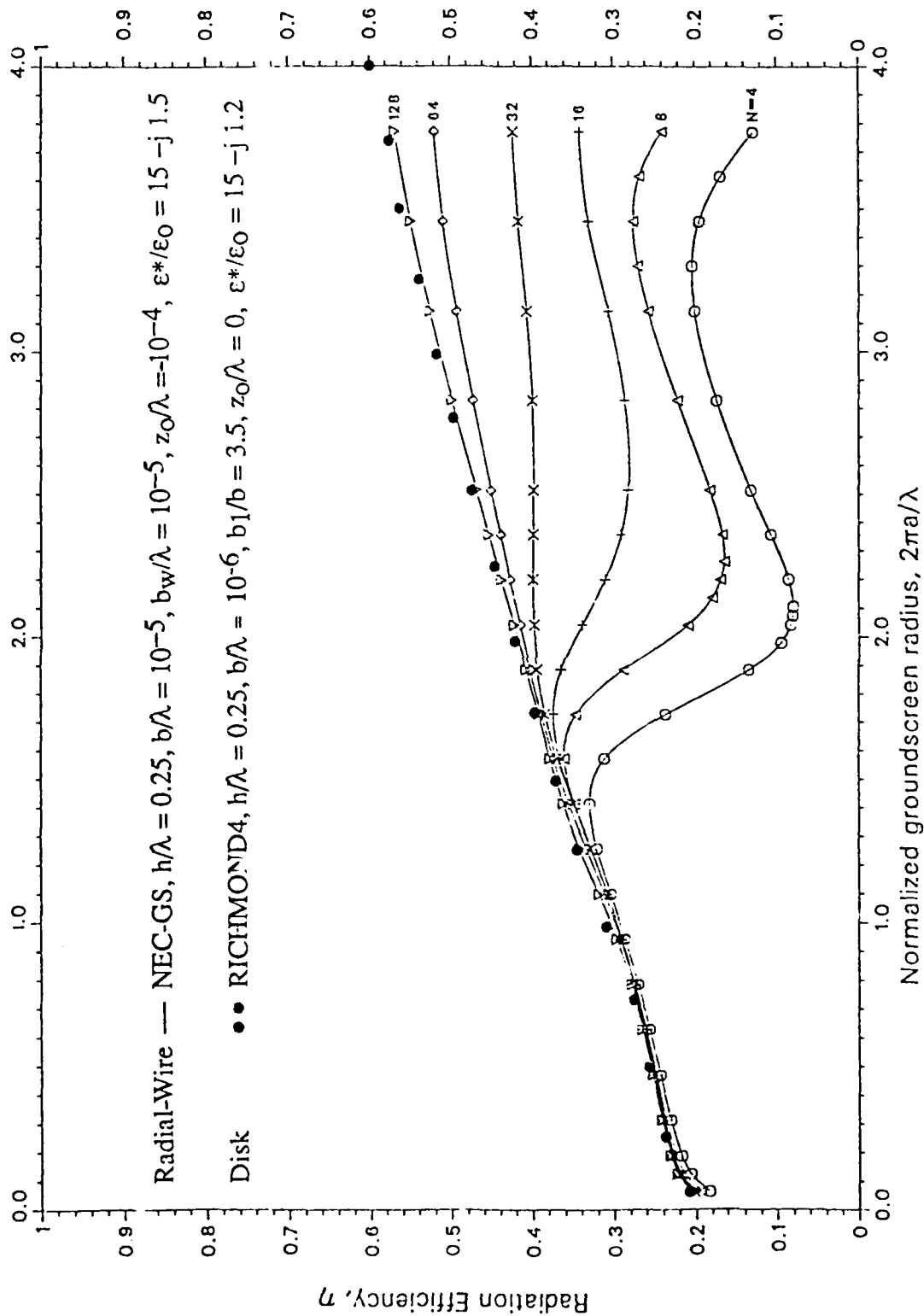


Figure 6. Radiation Efficiency of a Quarter-Wave Monopole Element with Radial Wire and Disk Ground Planes on or just above Medium Dry Earth

## SECTION 6

### CONCLUSIONS

This paper validates the NEC-3 and NEC-GS versions of the Numerical Electromagnetic Code for wire elements (vertical dipoles with no ground plane and monopoles with radial-wire ground planes) in close proximity to flat Earth.

The Fresnel reflection coefficient option of NEC-3 yields poor results for input current, input impedance, and radiation efficiency. Correct results for directivity are obtained for the case of an element (with no ground plane) in proximity to earth.

The NEC-3 Sommerfeld integral option, with its NEC-GS version, is probably the best available model for monopole elements with radial-wire ground planes (just as Richmond's method-of-moments program is the best available model for monopole elements with disk ground planes) provided that the ground planes are not so large that the maximum matrix size of the program is exceeded or that the computer run time is too excessive. Inaccuracies as much as 25 percent or more occur in computing input current, input impedance, and radiation efficiency for antenna elements in close proximity to lossy earth.

Version NEC-GS is a more efficient version of the NEC-3 Sommerfeld integral option for wire antennas that have rotational symmetry in the azimuthal direction. The NEC-GS rotational model gives close agreement with NEC-3 when the total surface area of the rotational elements is equal to the surface area of the physical element. When this condition is not satisfied, inaccuracies of ten percent or more can occur in input current, input impedance, and radiation efficiency. The format for the field of the input parameters is somewhat user-unfriendly because the omission of a concluding comma in the GR card increases the number of radials by a factor of ten.

### ACKNOWLEDGMENTS

I would like to thank G. J. Burke of Lawrence Livermore National Laboratory (LLNL) who made the initial comparisons shown in Figures 1 and 2 and Table 1; and L. Giandomenico who performed the NEC computer runs. S. Lamoureux typed the manuscript.

### LIST OF REFERENCES

1. Burke, G. J. and A. J. Poggio, January 1981, "Numerical Electromagnetics Code (NEC) — Method of Moments," Lawrence Livermore National Laboratory, Report UCID-18834.
2. Burke, G. J. and E. K. Miller, 1984, "Modeling Antennas Near to and Penetrating a Lossy Interface," *IEEE Trans. Antennas and Propagation*, Vol. AP-32, pp. 1040-1049.
3. E. K. Miller, A. J. Poggio, G. J. Burke and E. S. Selden, 1972, "Analysis of wire antennas in the presence of a conducting half space: Part I, The Vertical Antenna in Free Space," *Canadian J. Phys.*, vol. 50, pp. 879-888.

## LIST OF REFERENCES (CONTINUED)

4. Miller, E. K., A. J. Poggio, G. J. Burke and E. S. Selden, 1972, "Analysis of wire antennas in the presence of a conducting half space: Part II, The Horizontal Antenna in Free space," *Canadian J. Phys.*, vol. 50, pp. 2614-2627.
5. Wait, J. R., and W. J. Surtees, May 1954, "Impedance of a Top-Loaded Antenna of Arbitrary Length Over a Circular Grounded Screen," *Journal of Applied Physics*, Vol. 25, pp. 553-556.
6. Burke, G. J., February 1986, "Modeling Monopoles on Radial-Wire Ground Screens", *Applied Computational Electromagnetics Newsletter*, Vol. 1, No. 1.
7. Burke, G. J., June 1991, *User's Guide Supplement for NEC-GS*, Lawrence Livermore National Laboratory, Report UCRL-MA-107572.
8. Burke, G. J., January 1988, "A Model for Insulated Wires in the Method of Moments Code NEC," Lawrence Livermore National Laboratory, Report UCID-21301.
9. Private Communication from G. J. Burke, January 15, 1991, Lawrence Livermore National Laboratory.
10. Burke, G. J., 1989, "Validation of NEC for Antennas Near Ground," Lawrence Livermore National Laboratory, Report UCRL-101190. Also available from National Technical Information Service as DE89-016502.
11. CCIR (1986), "Electrical Characteristics of the Surface of the Earth," Recommendation 527-1 (Based on CCIR Report 229, 1982), 16th Plenary Assembly, Dubrovnik (1986), Int. Radio Consultative Committee (CCIR), Int. Telecommunication Union, Geneva, Switzerland, 1986.
12. King, R. W. P., 1971, *Tables of Antenna Characteristics*, IFI/Plenum, pp. 1-5. Please note in reference [12] that the complex wave number  $k = -j\gamma$  where  $\gamma = \alpha + j\beta$  is the complex propagation constant.
13. Adler, R. B., L. J. Chu, and R. M. Fano, 1960, *Electromagnetic Energy Transmission and Radiation*, New York: John Wiley and Sons, pp. 180-185, 204-207.
14. Stratton, J. A., *Electromagnetic Theory*, 1941, New York: McGraw-Hill, p. 494.
15. Reed, H. R., and C. M. Russell, 1964, *Ultra-High Frequency Propagation*, Lexington MA: Boston Technical Publishers, p. 90.
16. Burke, G. J., Fall 1988, "EM Modeling Notes," *Applied Computational Electromagnetics Society Journal and Newsletter*, Vol. 3, No. 3, p. 27.
17. Weiner, M. M., September 1991, *Performance of Ground-Based, High-Frequency, Receiving Arrays with Electrically-Small Ground Planes*, MITRE Technical Report MTR-11277, The MITRE Corporation, Bedford, MA. Also available from National Technical Information Service as AD-A250489. Accepted for publication in *Progress in Electromagnetics Research*.

## LIST OF REFERENCES (CONTINUED)

18. King, R. W. P., 1991, "On the Radiation Efficiency and the Electromagnetic Field of a Vertical Electric Dipole in the Air Above a Dielectric or Conducting Half-Space," *Progress in Electromagnetics Research (PIER 4)*, 1991, New York: Elsevier, Chapter 1.
19. Richmond, J. H., June 1985, "Monopole Antenna on Circular Disk Over Flat Earth," *IEEE Trans. Antennas and Propagation*, Vol. AP-33, No. 6, pp. 633-637
20. Private communication from G. J. Burke, January 7, 1991, Lawrence Livermore National Laboratory.
21. Weiner, M. M., June 1992, "Far-Zone Field of a Monopole Element on a Disk Ground Plane Above Flat Earth," MTR-92B0000090, The MITRE Corporation, Bedford, MA. Also available from National Technical Information Service as AD-A253580.
22. Private communication from G. J. Burke, April 1, 1992, of Lawrence Livermore National Laboratory.
23. Rama Rao, B., and D. N. Jones, June 24-28, 1991, "Characterization of a High-Frequency Beverage Antenna Using a Fiber Optic Measurement Technique," Digest, Vol. 2, International IEEE/AP-S Symposium, London, Ontario, pp. 1190-1193.
24. King, R. W. P., November 1983, "The Wave Antenna for Transmission and Reception", *IEEE Trans. Antennas and Propagation*, Vol. AP-31, pp. 956-965.
25. Op. cit. 1, Part III, pp. 66-68.
26. Private communication from G. J. Burke, November 21, 1990, of Lawrence Livermore National Laboratory,
27. Ludwig, A., September 1987, "Wire Grid Modeling of Surfaces," *IEEE Trans. Antennas and Propagation*, Vol. AP-35, No. 9, pp. 1045-1048.
28. Paknys, R. J., July 1991, "The Near Field of a Wire Grid Model," *IEEE Trans. Antennas and Propagation*, Vol. AP-39, No. 7, pp. 994-999.
29. Burke, G. J., October 1983, *User's Guide Supplement for NEC-3 for Modeling Buried Wires*, Lawrence Livermore Laboratory, Report UCID-19918.
30. Burke, G. J., March 1992, "A Comparison of Solutions for Wires Over Grounds," *Conference Proceedings, 8th Annual Review of Progress in Applied Computational Electromagnetics*, Naval Postgraduate School, Monterey, CA, March 16-20, 1992.
31. Harnish, L. O., M. Lee, and G. H. Hagn, March 1987, "Comparison of Measured and NEC - Calculated Characteristics of a Vertical Monopole with Buried Ground Radials," *Conference Proceedings 3rd Annual Review of Progress in Applied Computational Electromagnetics*, Naval Postgraduate School, Monterey, CA, March 24-26, 1987
32. Hagn, G. H. , June 1988, "HF Receiving Antenna Directivity Patterns and Gain for Ionospheric Propagation Model Predictions for Shortwave Broadcasting," *IEEE Trans. Broadcasting Technology*, Volume BT-34, No. 2, pp. 221-229.

### LIST OF REFERENCES (CONCLUDED)

33. Weiner, M. M. and S. Zamosciany, September 1991, *Radiation Efficiency and Input Impedance of Monopole Elements with Radial-Wire Ground Planes in Proximity to Earth*, MITRE Paper M91-104, The MITRE Corporation, Bedford, MA, Figures 2 and 6. Also available from National Technical Information Service as AD A244578.
34. Weiner, M. M., Zamosciany, S., and G. J. Burke, 30 July 1992, "Radiation Efficiency and Input Impedance of Monopole Elements with Radial-Wire Ground Planes in Proximity to Earth," *Electronics Letters*, Vol. 28, No. 16, pp. 1550-1551.
35. Weiner, M. M., 3 December 1992, "Radiation Efficiency and Directivity of Monopole Elements with Disc Ground Planes on Flat Earth," *Electronics Letters*, Vol. 28, No. 25, pp. 2282-2283.

**PROPAGATION OF VLF RADIATION IN THE EARTH-IONOSPHERE  
WAVEGUIDE EXCITED BY AN AIRBORNE DUAL  
TRAILING WIRE ANTENNA**

**D. H. Werner**

**The Applied Research Laboratory  
The Pennsylvania State University  
P. O. Box 30  
State College, PA 16804**

**J. K. Breakall**

**Electrical and Computer Engineering  
The Pennsylvania State University  
University Park, PA 16802**

**R. J. Lunnen**

**The Applied Research Laboratory  
The Pennsylvania State University  
P. O. Box 30  
State College, PA 16804**

*Abstract* - Field strength variations produced by an orbiting aircraft dual trailing wire VLF transmitting antenna are investigated. The towplane is assumed to be executing a circular orbit at a constant altitude and speed. A steady-state mechanical model is adopted for determination of the shape of the dual trailing wire antenna. The exact current distribution on this antenna is calculated using the Numerical Electromagnetics Code (NEC) which is based on a method of moments solution of the Electric Field Integral Equation (EFIE). A propagation code developed at the Naval Ocean Systems Center (NOSC) called TWIRE has been modified to be used in conjunction with NEC. This modified version of TWIRE has been called TWIRENEC. The TWIRENEC code uses the current distribution information provided by NEC to determine the dipole moments for a segmented antenna. The wire segmentation geometry and corresponding dipole moments are then used to calculate the electric field strength as a function of distance and azimuth in the earth-ionosphere waveguide. The waveguide can be considered as either horizontally homogeneous or inhomogeneous. It is demonstrated that the periodic variations in field intensity resulting from an orbiting transmitter are a function of receiver position. These periodic variations can range from a small fraction of a dB to several dB depending upon the location of the receiver with respect to the transmitter. A point dipole approximation of the dual trailing wire antenna is suggested for use in the study of VLF radiation excited by an orbiting antenna in the presence of wind shear. The point dipole approximation is applied to estimate the field strength variations caused by a yo-yo oscillation of the transmitting antenna as it orbits. These yo-yo oscillations are characterized in terms of the change in verticality of the point dipole which occurs over one complete orbit.

## I. INTRODUCTION

Kelly [1] investigates the VLF field strength variations resulting from an elevated and inclined point dipole transmitting antenna travelling in a circular orbit. This antenna is an idealization of a trailing wire antenna carried by an orbiting aircraft. The ionosphere is considered to be homogeneous and isotropic in the earth-ionosphere waveguide propagation model used for calculating VLF fields. An excellent discussion is presented on additional complications which would be introduced by an anisotropic ionosphere.

Pappert and Bickel [2] present theoretical expressions for the vertical ( $E_z$ ) and horizontal ( $E_y$ ) VLF electric fields excited by point dipoles of arbitrary orientation and elevation. Bickel et al. [3] and Bickel [4] apply these results to determine the VLF fields produced by an arbitrarily shaped wire composed of a series of point dipole radiators, each with the proper orientation, elevation and current moment. The field strength at any distance away from the antenna was found by taking the vector sum of the contributions from each point dipole. This extended source model was used to calculate the amplitude of vertical and horizontal field components for various configurations of an airborne VLF trailing wire antenna. Predicted values of the vertical field component  $E_z$  for daytime and nighttime propagation were generally found to be in good agreement with measurements. Pappert and Hitney [5] describe a propagation code called TWIRE which streamlines calculations of the type made by Bickel et al. [3] and Bickel [4]. One of the improvements incorporated into the TWIRE code is the ability to determine the current amplitude from an assumed current distribution and radiated power.

A modified version of the TWIRE code, called TWIRENEC, has been created by the authors. The major difference between the two codes is in the way the current distribution on the VLF antenna is determined. The TWIRE code makes the assumption that the current distribution on the antenna is sinusoidal and relates the current amplitude to an assumed radiated power. The TWIRENEC code uses the exact current distribution calculated by the Numerical Electromagnetics Code (NEC) [6]. This current distribution is calculated using an appropriate value of input power delivered to the antenna by the transmitter. This paper presents and discusses some VLF propagation results obtained using the new TWIRENEC code. In particular, this code is used to compute field strength variations caused by an orbiting aircraft which is trailing a VLF transmitting antenna.

## II. THEORY

The TWIRENEC propagation code was developed for calculating VLF fields in the earth-ionosphere waveguide generated by antennas of arbitrary length, shape and elevation. A flowchart of the TWIRENEC computer program is shown in Figure 1. The original TWIRE subroutines COORD, RPOWER and FASTMC have been renamed COORD2, RPOWER2 and FASTMC2 to indicate that they were modified for use in TWIRENEC.

The subroutine COORD2 reads the wire segmentation geometry and the corresponding dipole moments from output files created by the NEC code. The wire segmentation geometry is then transformed from the towplane coordinate system to the

propagation coordinate system. Figure 2 is an illustration of the propagation coordinate system. The direction of propagation is along the positive  $x$  axis which is at a bearing  $\varphi_a$  with respect to magnetic north. The towplane is assumed to be orbiting at an altitude  $z = z_p$  with an orbital radius of  $r_p = \sqrt{x_p^2 + y_p^2}$  and a constant velocity  $\vec{V}$ . The position of the towplane in its orbit with respect to the propagation axis may be described in terms of the angle  $\psi$ . One complete orbit of the towplane is characterized by a progression in the angle  $\psi$  from  $0^\circ$  to  $360^\circ$ .

The subroutine RPOWER2 calculates the time averaged radiated power from the wire segmentation geometry and dipole moments which are output from COORD2. The expression for radiated power used in this subroutine was derived by Pappert [7] assuming thin antennas of arbitrary elevation and orientation above perfectly conducting ground. This formulation is based upon segmentation of the wire antenna.

The mode sums for a horizontally inhomogeneous waveguide excited by an antenna which is composed of  $W$  segments are [5]

$$E_n^{(p)}(\mu V/m) = Q \sum_j \sum_m a_{jm}^{(p)} \left[ \delta_{1n} \frac{S_j^{(p)}}{S_m^{(1)}} + (1 - \delta_{1n}) \right] f_{nj}^{(p)}(z_r) \quad (1)$$

$$\cdot \exp \left[ -ik(S_m^{(1)}x_2 + S_j^{(p)}(x - x_p)) \right] F_{nm}, p \neq 1$$

where

$$F_{nm} = \sum_{w=1}^W \frac{M_w \exp(ikS_m^{(1)}\bar{x}_w)}{\left[ \sin\left(\frac{x - \bar{x}_w}{a}\right) \right]^{1/2}} \left[ \lambda_{1nm}^{(1)} f_{1m}^{(1)}(\bar{z}_w) \cos\gamma_w \right. \quad (2)$$

$$\left. + \lambda_{2nm}^{(1)} f_{2m}^{(1)}(\bar{z}_w) \sin\gamma_w \sin\phi_w + \lambda_{3nm}^{(1)} f_{3m}^{(1)}(\bar{z}_w) \sin\gamma_w \cos\phi_w \right]$$

and

$Q$	=	$2.853 \times 10^{-3} f^{3/2}$
$S_m$	=	$\sin \theta_m$
$f$	=	frequency (kHz)
$k$	=	free space wave number
$a$	=	radius of the earth
$z_r$	=	receiver altitude
$w$	=	wire segment index
$(\bar{x}_w, \bar{z}_w)$	=	midpoint coordinates for the $w^{\text{th}}$ segment
$(\gamma_w, \phi_w)$	=	orientation angles for the $w^{\text{th}}$ segment
$M_w$	=	dipole moment of the $w^{\text{th}}$ segment (rms amp-meters)



$p$	=	waveguide slab number
$j, m$	=	mode indices
$\theta_m$	=	modal eigenangle
$a_{jm}^{(p)}$	=	cumulative mode conversion coefficients
$\delta_{1n}$	=	Kronecker delta function
$E_n$	=	electric field component at the receiver ( $n=1$ implies $E_z$ , $n=2$ implies $E_y$ and $n=3$ implies $E_x$ )
$i$	=	$\sqrt{-1}$
$f_n$	=	modal height gain ( $f_1$ is height gain for $E_z$ , $f_2$ is height gain for $E_y$ and $f_3$ is height gain for $E_x$ )
$\lambda_n$	=	excitation factor ( $\lambda_1$ is the vertical dipole excitation factor, $\lambda_2$ is the broadside dipole excitation factor and $\lambda_3$ is the end-on dipole excitation factor)

The wave propagation is assumed to be in the  $x$ - $z$  plane with the  $x$  coordinate as the range. The origin for the  $x$  coordinate is taken to be at the center of the towplane orbit projected onto the ground. The  $z$  coordinate has its origin on the ground and is directed positive towards the ionosphere. The midpoint of the  $w^{\text{th}}$  dipole with current moment  $\vec{M}_w$  is located at  $(\bar{x}_w, \bar{y}_w, \bar{z}_w)$  with orientation  $\phi_w$  and  $\gamma_w$  relative to the  $x$  and  $z$  axis, respectively. The  $a_{jm}^{(p)}$  represent cumulative mode conversion coefficients for a slab mode conversion model in which  $x_n$  is the beginning of the  $n^{\text{th}}$  slab and with the first slab described by the region  $x < x_2$ . The physical interpretation of the cumulative mode conversion coefficients are the accumulative conversion from a unit amplitude wave in mode  $m$  in the transmitter region to mode  $j$  in the  $p^{\text{th}}$  slab. The electromagnetic field strength in the waveguide is calculated along a propagation path by making use of the subroutine FASTMC2. FASTMC2 is essentially a fast mode conversion propagation code developed by Ferguson and Snyder [8]. The subroutines PRESEG, MODEFNDR and SEGMWVGD are part of the Naval Ocean System Center's Long-Wave Propagation Capability (LWPC) program. The LWPC code is documented in Ferguson and Snyder [9] and in Ferguson et al. [10]. PRESEG is a driver program which sets up files that provide the necessary input and calls to the subroutines MODEFNDR [11] and SEGMWVGD [12]. Data files are set up by PRESEG as input for MODEFNDR which obtains starting mode solutions for a specific segment on the propagation path. The starting solutions determined by MODEFNDR are input to SEGMWVGD. The SEGMWVGD program then extrapolates these solutions as the waveguide parameters vary with distance from the transmitter. The resulting horizontally inhomogeneous waveguide parameters are then input into the FASTMC2 program to be used in the calculation of mode conversion coefficients and mode sums. The programs PRESEG, MODEFNDR and SEGMWVGD can be used to find the waveguide parameters for a horizontally homogeneous as well as a horizontally inhomogeneous waveguide. The mode sums for the fields in a horizontally homogeneous waveguide ( $p=1$ ) excited by a W segmented antenna are

$$E_n^{(1)}(\mu V/m) = Q \sum_m f_{nm}^{(1)}(z_r) \exp[-ik S_m^{(1)} x] F_{nm} \quad (3)$$

A horizontally inhomogeneous waveguide model is more realistic than a model which is horizontally homogeneous. This is because a horizontally inhomogeneous model can take into account variations in the ground conductivity and ionosphere which may be present over a propagation path.

Bickel [4] and Pappert and Hitney [8] have shown that the trailing wire transmitting antenna can be approximated by a point dipole with the properly chosen current moment and orientation. In most cases the altitude and orientation of the point dipole can be chosen to correspond to the towline segment which contains the current maximum. The current on this point dipole is chosen so that it has a radiated power equivalent to that of the trailing wire antenna it is intended to replace. This is an important result because it implies that a considerable reduction in computation time can be achieved by replacing the complex structure of the trailing wire antenna with an appropriate point dipole.

One application where the point dipole approximation may be useful is in the study of propagation modes excited by a trailing wire which has periodic yo-yo oscillations. Yo-yo oscillations in the long trailing wire antenna frequently occur and have been verified by altitude measurements of a drogue located at the end of the wire [13]. In particular, oscillations in the drogue altitude of several thousands feet with a period equivalent to the orbital period have been observed. These yo-yo oscillations are believed to be the result of a variation in the wind velocity as a function of altitude, i.e. wind shear. The point dipole approximation suggests that a knowledge of the influence of yo-yo motion on the segment which contains the maximum current should be sufficient to characterize the transmitting antenna. Kelly [1] considers a model in which yo-yo produces a periodic variation in inclination of a point dipole antenna as it traverses a circle. The yo-yo motion is taken into account by allowing the inclination angle  $\gamma$  to oscillate about some average value  $\gamma_o$  as the antenna orbits. This periodic change in inclination can be represented by

$$\gamma = \gamma_o + \Delta\gamma \sin(\psi + \psi_o) \quad (4)$$

where  $\psi_o$  is an offset angle in the orbital period of the inclination. The variable  $\Delta\gamma$  determines the amplitude of the yo-yo induced excursions in the inclination. When  $\Delta\gamma = 0$  there are no oscillations in inclination which indicates that static conditions exist.

A reasonable assumption to make is that  $0^\circ < \gamma_o < 90^\circ$  for the segment of the trailing wire antenna which has the current maximum. A typical value of  $\gamma_o$  would be  $45^\circ$ . It is physically realistic to assume that the change in inclination due to yo-yo must have the vertical and horizontal as the two limiting positions of the point dipole, hence  $0^\circ \leq \gamma \leq 90^\circ$ . In order to satisfy these conditions,  $\Delta\gamma$  must be restricted to lie within the range given by

$$0^\circ \leq \Delta \gamma \leq \Delta \gamma_{\max} = \min (\gamma_o, 90^\circ - \gamma_o) \quad (5)$$

At this point in the development it is convenient to introduce the concept of verticality. Verticality is defined to be the ratio of the vertical projection of the wire antenna to the total length of the antenna, in percent. For example, consider a point dipole antenna which is inclined at an angle  $\gamma_o$  from the vertical. The verticality  $V_o$  of this antenna is then

$$V_o = 100 \cos \gamma_o \quad \% \quad (6)$$

The verticality would be about 70% for a point dipole inclined at an angle of  $45^\circ$  with respect to the vertical.

The minimum and maximum angle of inclination which result from yo-yo motion are

$$\gamma_{\min} = \gamma_o - \Delta \gamma \quad (7)$$

$$\gamma_{\max} = \gamma_o + \Delta \gamma \quad (8)$$

The corresponding minimum and maximum verticalities are

$$V_{\min} = 100 \cos \gamma_{\max} \quad \% \quad (9)$$

$$V_{\max} = 100 \cos \gamma_{\min} \quad \% \quad (10)$$

The change in verticality over one complete orbit is then

$$\Delta V = V_{\max} - V_{\min} \quad \% \quad (11)$$

Equations (7) - (11) can be used to derive an expression which relates the yo-yo oscillation amplitude  $\Delta \gamma$  to the change in verticality  $\Delta V$ . This relationship is

$$\Delta \gamma = \sin^{-1} \left[ \frac{\Delta V(\%)}{200 \sin \gamma_o} \right] \quad (12)$$

The bounds on  $\Delta V$  can be obtained by substituting (12) into (5), which results in

$$0 \leq \Delta V \leq \Delta V_{\max} = 200 \sin \gamma_o \min (\sin \gamma_o, \cos \gamma_o) \quad (13)$$

### III. RESULTS

Propagation characteristics of the 22 kHz dual trailing wire antenna shown in Figure 3 will be investigated in this section. Steady-state mechanical modeling codes, which are based on an analysis by Huang [14] and Narkis [15], were used to arrive at the geometry of this antenna. Projections of the antenna geometry onto the three principal planes along with the orbital path are displayed in this figure. Under steady state conditions, the antenna is assumed to maintain the same shape as it is being trailed by the orbiting towplane. The towplane trailing the antenna configuration shown in Figure 3 is orbiting counterclockwise. The length of the short and long trailing wires are 2680 feet and 19500 feet, respectively, which corresponds to an electrical length of one-half wavelength ( $\lambda/2$ ). The altitude of the towplane is 20500 feet, the orbit radius is 6073 feet and its speed is 199 knots. The verticality of the short wire is 30.66% and the verticality of the long wire is 69.81%. The field strength variations produced by this antenna as it moves around in a circle are of particular interest. The current distribution along this antenna was determined using the NEC code and assuming a 200 kW input power with a DC wire resistance of 4.5  $\Omega$ /1000 feet. The presence of the aircraft was neglected in the modeling of the dual trailing wire antenna system.

An easterly propagation path ( $\phi_a = 90^\circ$ ) over seawater is assumed in the analysis presented in this section. The values used for conductivity and relative permittivity of seawater were 4.64 S/m and 81, respectively. An exponential electron density profile of the form [16]

$$N(z) = 1.46 \times 10^7 \exp[(\beta - 0.15)z - \beta h'] \text{ cm}^{-3} \quad (14)$$

was adopted. The parameters of this model were chosen as  $\beta = 0.5 \text{ km}^{-1}$  and  $h' = 87 \text{ km}$ . This particular electron density profile has been shown to accurately predict nocturnal VLF propagation to the east [17]. The geomagnetic field strength was taken to be 0.5 Gauss with a dip angle of  $50^\circ$ .

Figure 4 shows curves of the vertical electric field component  $E_z$  at sea level expressed in dB above  $1 \mu\text{V/m}$ . The two curves shown in Figure 4 compare nocturnal easterly VLF propagation at 22 kHz for an orbiting towplane at positions which are  $90^\circ$  (solid curve) and  $270^\circ$  (dashed curve) with respect to the direction of propagation. A relatively large difference between the two signals is observed at a distance of 1.7 Mm, whereas a relatively small difference exists at 3.8 Mm. Figure 5 shows the orbital dependence of the  $E_z$  field component amplitude at a range of 1.7 Mm (diamonds) and 3.8 Mm (triangles) from the center of the towplane orbit. This demonstrates that it is possible to get a considerable variation in the signal level at the receiver caused by the orbiting of the towplane, in this case about 8 dB at 1.7 Mm. It is also possible, however, to select a suitable receiver location for which the variations in signal level due to orbiting are minimized (on the order of a few tenths of a dB at 3.8 Mm).

Figure 6 shows a plot of the electric field component  $E_z$  as a function of distance for an orbital position of  $0^\circ$  with respect to the direction of propagation (solid curve). The dashed curve represents a point dipole approximation to the dual trailing wire antenna. The point dipole was chosen to have the same altitude (4.976 km) and orientation ( $\gamma = 41.164^\circ$ ,  $\phi = -36.695^\circ$ ) as the segment containing the current maximum. Equation (6) can be used to show that the verticality of this segment is around 75%. The point dipole current was determined by requiring that the point dipole have the same radiated power as the trailing wire antenna. The radiated power of the trailing wire antenna shown in Figure 3 was calculated as 96.28 kW.

It was found that the resultant radiation fields in the waveguide are relatively insensitive to the shape of the short trailing wire. This can be attributed to the fact that the current which exists on the short wire is small in comparison to the current on the long wire. Figure 6 demonstrates that reasonably accurate agreement can be obtained by replacing the entire trailing wire antenna by a point dipole.

The point dipole approximation can be utilized to greatly simplify the study of field strength variations caused by wind induced yo-yo oscillations in the trailing wire. For instance, suppose that it is desired to ascertain the influence of yo-yo motion on the field intensity of the dual trailing wire antenna shown in Figure 3. This can be accomplished by using the point dipole approximation of this antenna (see Figure 6) to represent the steady-state orientation for the point dipole yo-yo model described in the previous section. Table 1 lists the values of several parameters associated with this point dipole yo-yo model. The parameters are calculated based upon an assumed change in verticality of 10%, 15%, 20%, 25% and 30%.

TABLE 1. POINT DIPOLE YO-YO MODEL PARAMETERS

$\gamma_o$ ( $^\circ$ )	$\Delta V_{\max}$ (%)	$\Delta V$ (%)	$\Delta \gamma_{\max}$ ( $^\circ$ )	$\Delta \gamma$ ( $^\circ$ )	$V_o$ (%)	$V_{\max}$ (%)	$V_{\min}$ (%)	$\gamma_{\max}$ ( $^\circ$ )	$\gamma_{\min}$ ( $^\circ$ )
41.164	86.650	10	41.164	4.357	75.283	80.065	70.065	45.521	36.807
41.164	86.650	15	41.164	6.543	75.283	82.292	67.292	47.707	34.621
41.164	86.650	20	41.164	8.739	75.283	84.409	64.409	49.903	32.425
41.164	86.650	25	41.164	10.947	75.283	86.413	61.413	52.111	30.217
41.164	86.650	30	41.164	13.173	75.283	88.302	58.302	54.337	27.991

Figure 7 contains plots of the vertical field strength as a function of distance for the steady-state case in which there would be a 0% change in verticality over one complete orbit of the towplane. The solid and dashed curves represent towplane angles of  $90^\circ$  and  $270^\circ$  in relation to the direction of propagation, respectively. The point dipole results of Figure 7 compare favorably with the dual trailing wire results shown in Figure 4. Since there is no change in verticality in this case (steady-state), the differences observed in the vertical electric field component are strictly a consequence of orbiting.

The yo-yo model of (4) with  $\psi_0 = 0^\circ$  can be used to show that a towplane angle of  $90^\circ$  with respect to the propagation direction corresponds to the point in the orbit where the verticality is the lowest. On the other hand, the verticality is highest when the towplane angle is  $270^\circ$ . The signal levels associated with the positions in the orbit for which the antenna attains the highest and the lowest verticality are compared in Figure 8 for  $\Delta V = 15\%$  and in Figure 9 for  $\Delta V = 30\%$ . Figure 9 suggests that an orbital verticality change of 30% would produce a significant variation in the vertical field intensity over nearly the entire propagation path. Figure 10 shows the yo-yo dependence of  $E_z$  at 2.6 Mm which results from an antenna that undergoes a 0% (circles), 15% (triangles) and 30% (diamonds) change in verticality as it orbits. Figure 11 shows the corresponding yo-yo dependence of  $E_z$  at 2.9 Mm, a point which is 300 km distant from 2.6 Mm. Figure 10 suggests that a receiver located at 2.6 Mm would see very little variation in the vertical field intensity under steady-state conditions. However, yo-yo motion of the antenna characterized by a 15% and a 30% change in verticality would result in field intensity variations on the order of 1.9 dB and 3.7 dB, respectively. Figure 11 indicates that, in the absence of yo-yo, the receiver would see a variation in field intensity of about 4.8 dB. With the introduction of yo-yo, however, the magnitude of these field strength variations increase. A 9 dB variation in field strength results from an orbital verticality change of 30%. Figure 11 demonstrates that the steady-state orbital dependence of the field intensity can be considerably magnified by dynamic yo-yo motion.

#### IV. SUMMARY

The field strength variations associated with an orbiting aircraft which is trailing a VLF transmitting wire antenna have been investigated in this paper. A steady-state mechanical modeling code was used to determine the wire shape geometry of the orbiting VLF antenna. The mechanical modeling code provided piecewise wire segmentation for data input to NEC. This allowed the exact current distribution on the antenna to be calculated by NEC. Finally, the output NEC current distribution was used by the TWIRENEC VLF propagation code to compute the electric field strength as a function of distance, azimuth and altitude in the earth-ionosphere waveguide.

A detailed discussion of the TWIRENEC code was presented in Section 2 of this paper. The major advantage of this code is that, for a specified antenna input power, the exact current distribution on the wire is used in the calculation of the radiated power and associated mode sums. Propagation over paths in which the ionospheric or ground parameters significantly vary is modeled by considering the earth-ionosphere waveguide to be horizontally inhomogeneous. In this paper, the TWIRENEC code was used to model

nocturnal easterly propagation over an all seawater path at a frequency of 22 kHz. Under these circumstances, a horizontally homogeneous waveguide model was sufficient to characterize the VLF propagation.

It was demonstrated that the entire segmented wire antenna can be approximated by a point dipole with an altitude and orientation chosen to correspond to the segment which contains the current maximum. A considerable reduction in TWIRENEC computation time was achieved by using the point dipole approximation of the trailing wire antenna. This paper combines the point dipole approximation with a yo-yo model in order to expedite the study of field strength variations caused by wind induced oscillations in the trailing wire. It was found that the steady-state orbital field strength variations tend to be localized, while the variations caused by yo-yo motion of the antenna are global in character. Results indicate that a magnification in the steady-state orbital dependence of the field intensity can be attributed to a yo-yo motion of the antenna.

### ACKNOWLEDGEMENTS

Discussions regarding the TWIRE VLF propagation code with K. Grauer and R. Pappert of the Naval Ocean Systems Center are gratefully acknowledged. The authors wish to thank J. A. Huffman for her assistance in computer programming. This work was supported by the TACAMO Aircraft Programs Office, PMA 271.

### REFERENCES

- [1] F. J. Kelly, "VLF field strength variations from an airborne trailing-wire antenna," *Radio Sci.*, 5 (5), 785-791, May 1970.
- [2] R. A. Pappert and J. E. Bickel, "Vertical and horizontal VLF fields excited by dipoles of arbitrary orientation and elevation," *Radio Sci.*, 5, 1445-1452, Dec. 1970.
- [3] J. E. Bickel, D. G. Morfitt, I. J. Rothmuller and W. F. Moler, Propagation Analysis of Diversity for VLF Communication Systems, Naval Electronics Laboratory Center Technical Document No. 139, Sept. 1971.
- [4] J. E. Bickel, Measurements of Vertical and Horizontal VLF Fields Excited by an Elevated, Arbitrarily Oriented Antenna, Naval Electronics Laboratory Center Technical Report No. 1833, July 1972.
- [5] R. A. Pappert and L. R. Hitney, "TWIRE:" A Test Program for VLF Airborne Transmissions, Naval Ocean Systems Center Technical Document No. 1499, April 1989.
- [6] G. J. Burke and A. J. Poggio, Numerical Electromagnetics Code (NEC) - Method of Moments, Naval Ocean Systems Center Technical Document No. 116, prepared by Lawrence Livermore Laboratory for the Naval Ocean Systems Center and Air Force Weapons Laboratory, July 1977.

- [7] R. A. Pappert, "Radiation resistance of thin antennas of arbitrary elevation and configuration over perfectly conducting ground," *IEEE Trans. Antennas Propagat.*, AP-35 (7), 863-866, July 1987.
- [8] J. A. Ferguson and F. P. Snyder, Approximate VLF/LF Waveguide Mode Conversion Model: Computer Applications: FASTMC and BUMP, Naval Ocean Systems Center Technical Document No. 400, Nov. 1980.
- [9] J. A. Ferguson and F. P. Snyder, Long-Wave Propagation Capability Program Description and User's Guide, Naval Ocean Systems Center Technical Document No. 1449, January 1989.
- [10] J. A. Ferguson, F. P. Snyder, D. G. Morfitt and C. H. Shellman, Software Program: Long-Wave Propagation Capability and Documentation, Naval Ocean Systems Center Technical Document No. 1518, March 1989.
- [11] C. H. Shellman, A New Version of MODESRCH Using Interpolated Values of the Magnetoionic Reflection Coefficients, Naval Ocean Systems Center Technical Report No. 1473, Oct. 1986.
- [12] J. A. Ferguson and F. P. Snyder, The Segmented Waveguide Program for Long-Wavelength Propagation Calculations, Naval Ocean Systems Center Technical Document No. 1071, Dec. 1986.
- [13] D. Nathans, TACAMO Dual Trailing Wire Antenna Input Impedance Measurements and Antenna Simulator Design, Naval Air Development Center Report No. 87040-40, Warminster, Pennsylvania, Dec. 1986.
- [14] S. L. Huang, Mathematical Model for Long Cable Towed by Orbiting Aircraft, Naval Air Development Center Report No. NADC-AM-6849, Warminster, Pennsylvania, June 1969.
- [15] Y. Narkis, "Approximate solution for the shape of flexible towing cables," *Journal of Aircraft*, 14 (9), 923-925, Sept. 1977.
- [16] J. R. Wait and K. P. Spies, Characteristics of the Earth-Ionosphere Waveguide for VLF Radio Waves, National Bureau of Standards Technical Note No. 300, Gaithersburg, Maryland, 1964.
- [17] R. A. Pappert and L. R. Hitney, "Empirical modeling of nighttime easterly and westerly VLF propagation in the earth-ionosphere waveguide," *Radio Sci.* 23 (4), 599-611, July 1988.



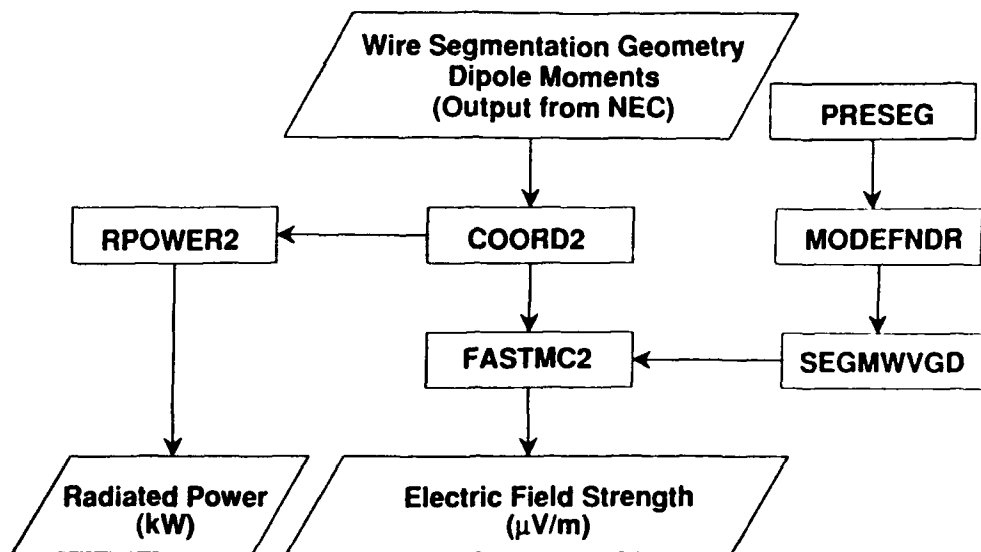


Figure 1. Flowchart of the TWIRENEC program.

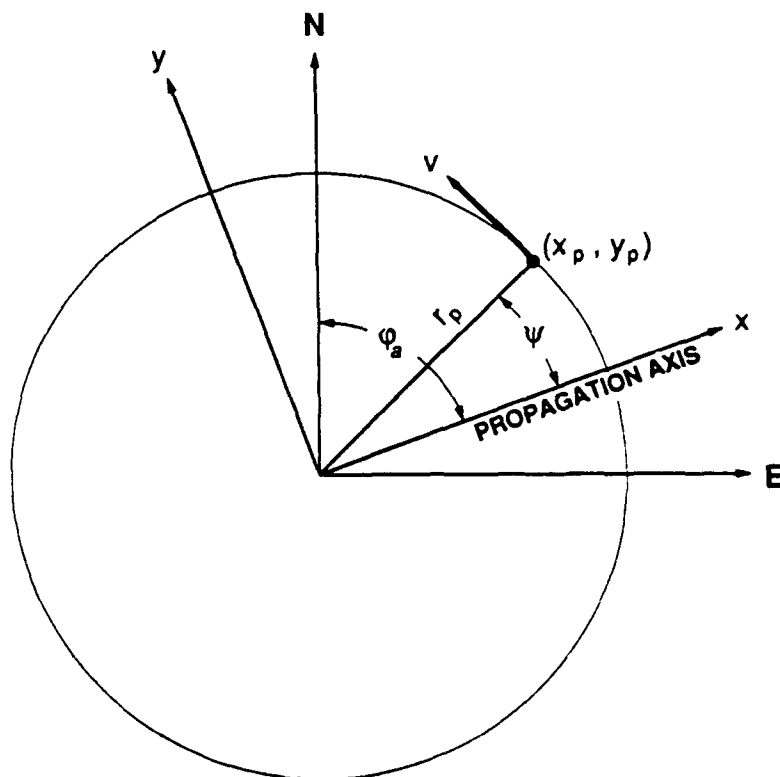


Figure 2. Propagation coordinate system.

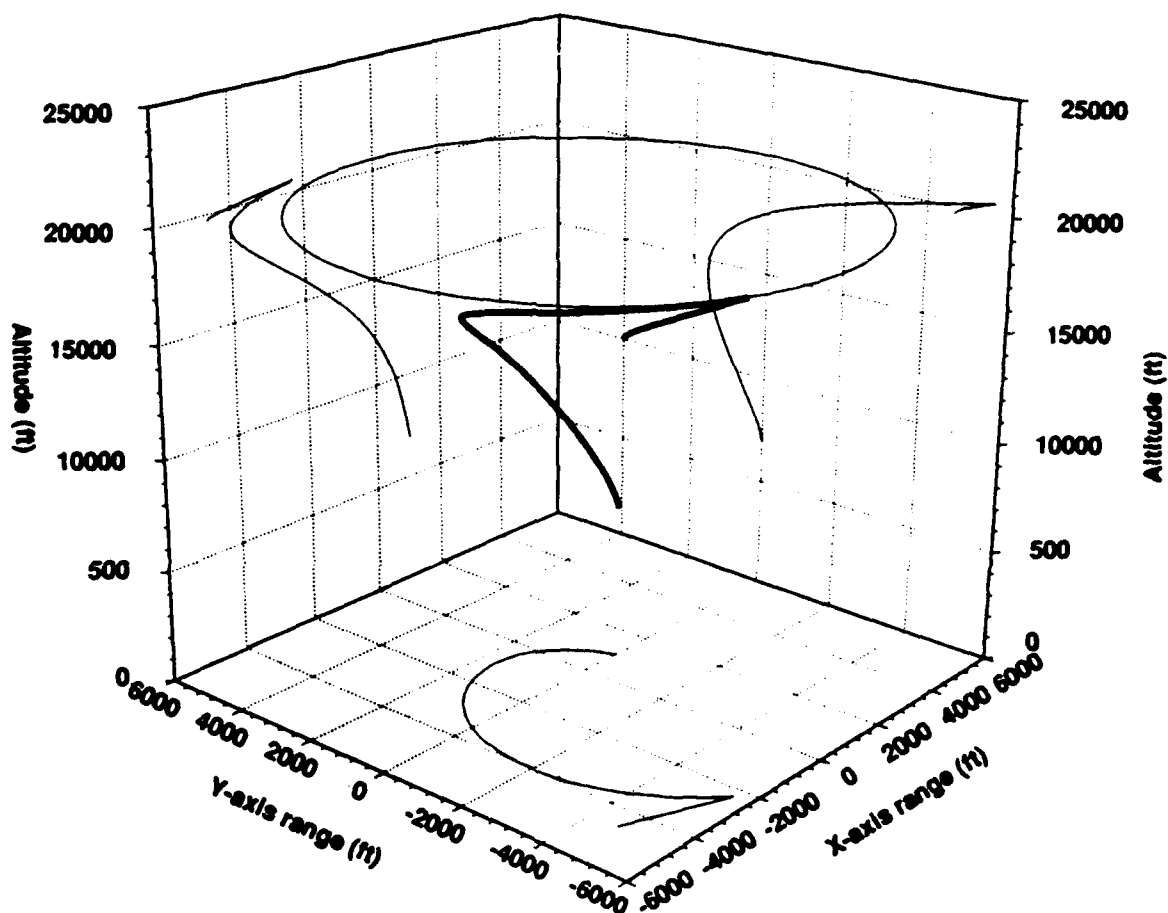


Figure 3. Dual trailing wire antenna steady-state geometry.

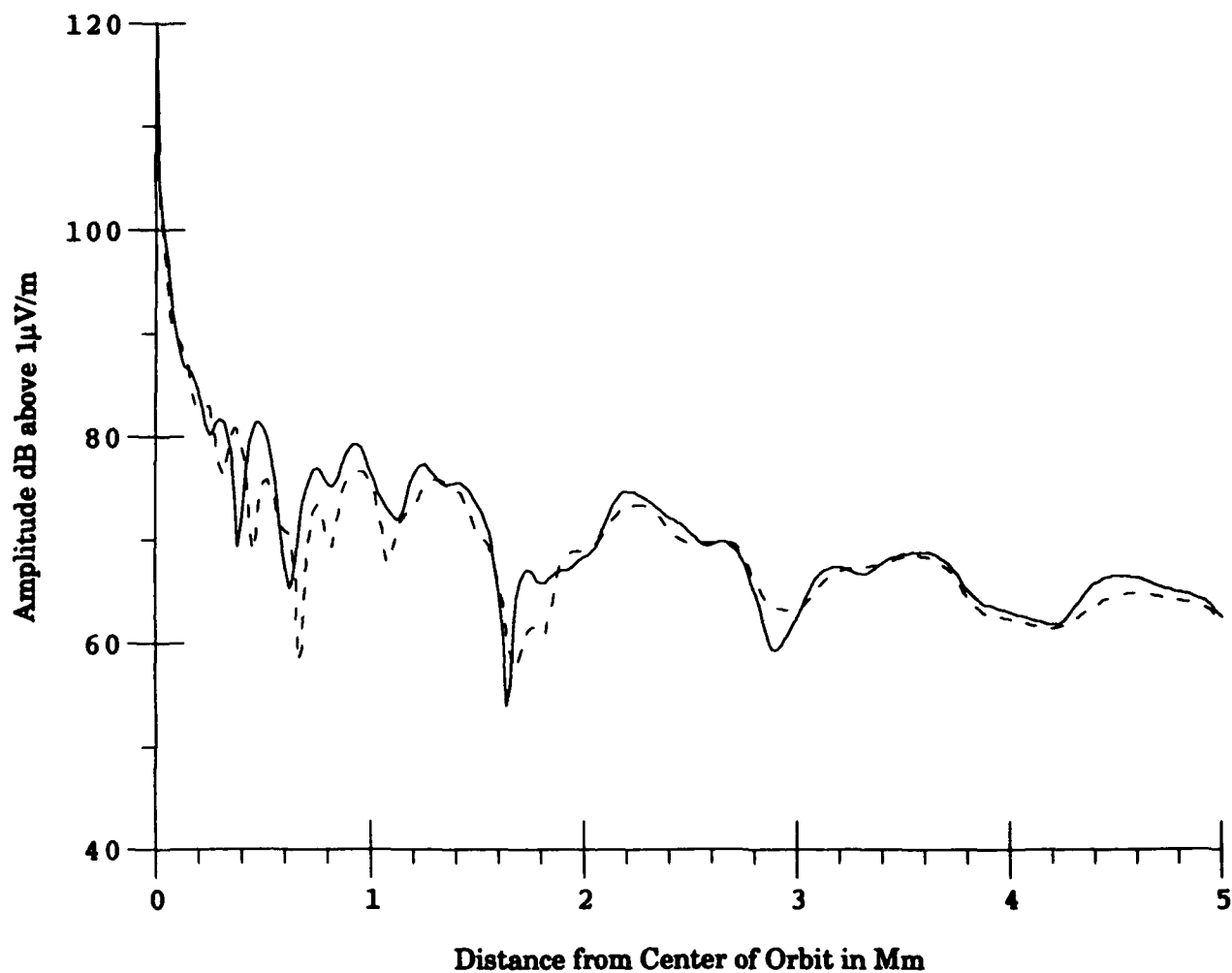


Figure 4. The vertical field strength  $E_z$  at sea level resulting from an orbiting towplane with  $\psi = 90^\circ$  (solid curve) and  $\psi = 270^\circ$  (dashed curve). Nocturnal easterly 22 kHz propagation from the dual trailing wire antenna.

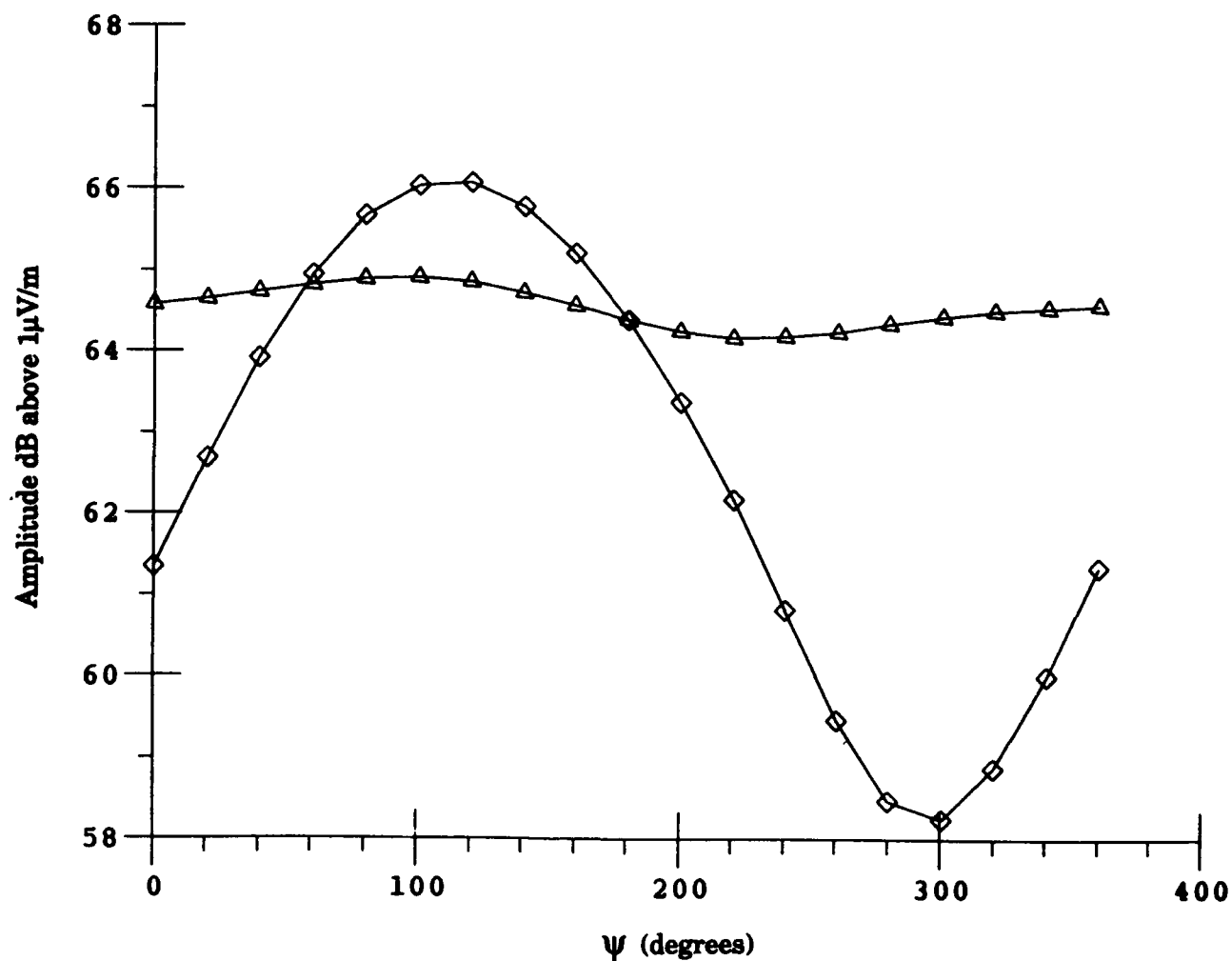


Figure 5. A comparison of the orbital dependence of the vertical field strength  $E_z$  at 1.7 Mm (diamonds) and at 3.8 Mm (triangles). Nocturnal easterly 22 kHz propagation from the dual trailing wire antenna.

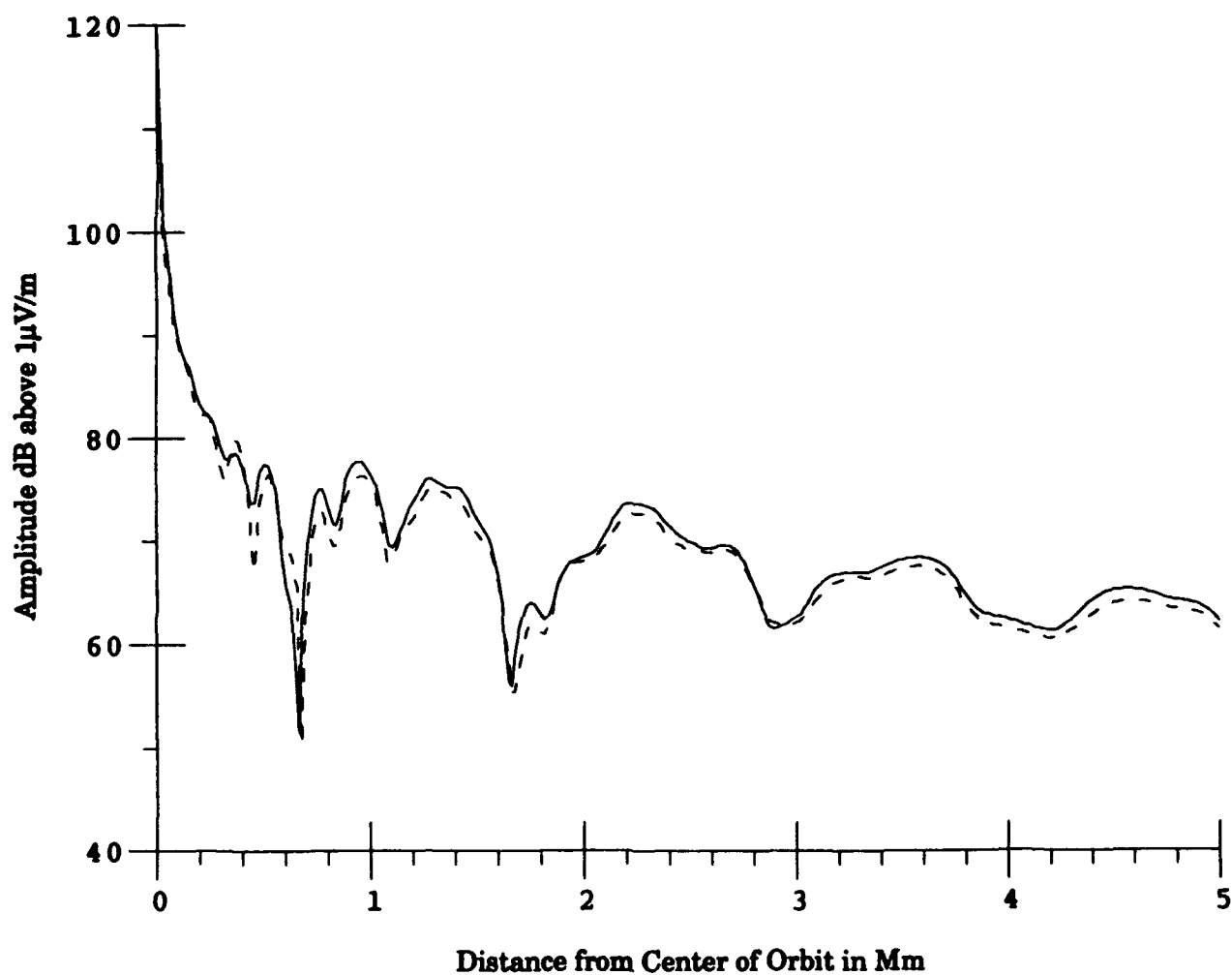


Figure 6. The vertical field strength  $E_z$  at sea level resulting from a dual trailing wire (solid curve) compared to a point dipole approximation (dashed curve). Nocturnal easterly 22 kHz propagation.

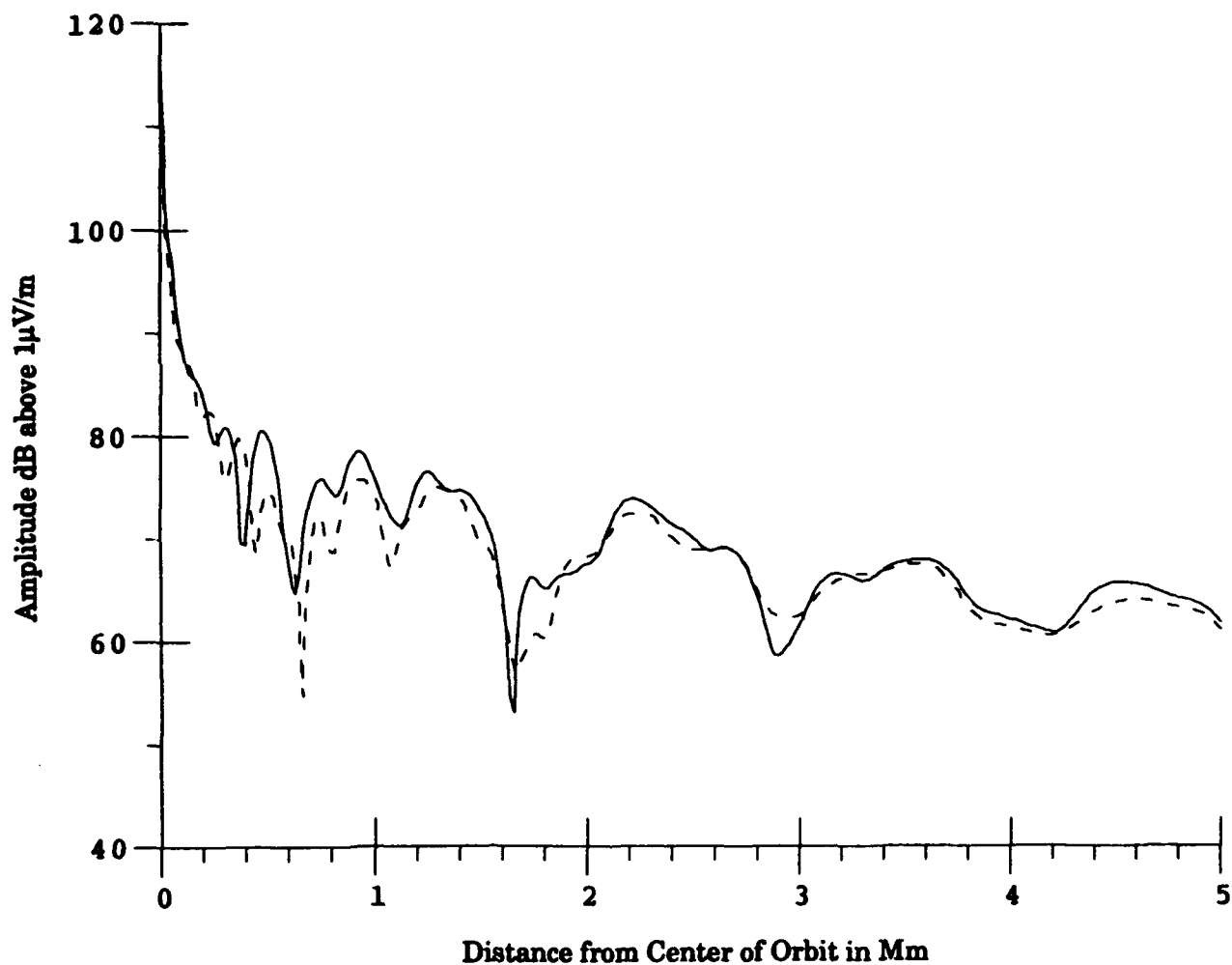


Figure 7. Vertical field strength  $E_z$  at sea level as a function of distance for the steady-state case ( $\Delta V = 0\%$ ). The solid and the dashed curves correspond to  $\psi = 90^\circ$  and  $\psi = 270^\circ$ , respectively. Nocturnal easterly 22 kHz propagation from a point dipole approximation of the dual trailing wire antenna.

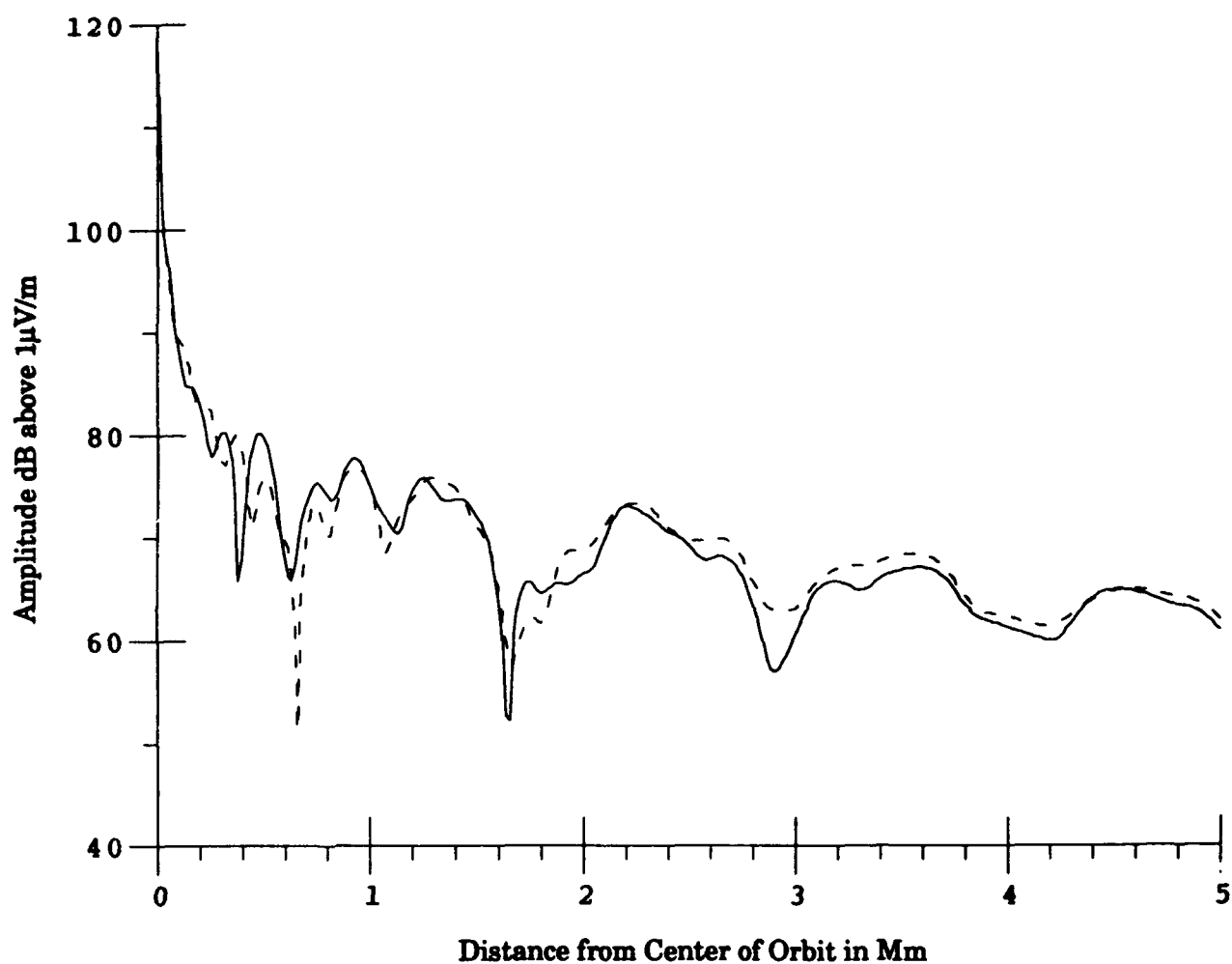


Figure 8. Vertical field strength  $E_z$  at sea level as a function of distance associated with  $\Delta V = 15\%$ . The solid and the dashed curves correspond to  $\psi = 90^\circ$  and  $\psi = 270^\circ$ , respectively. Nocturnal easterly 22 kHz propagation from a point dipole approximation of the dual trailing wire antenna.

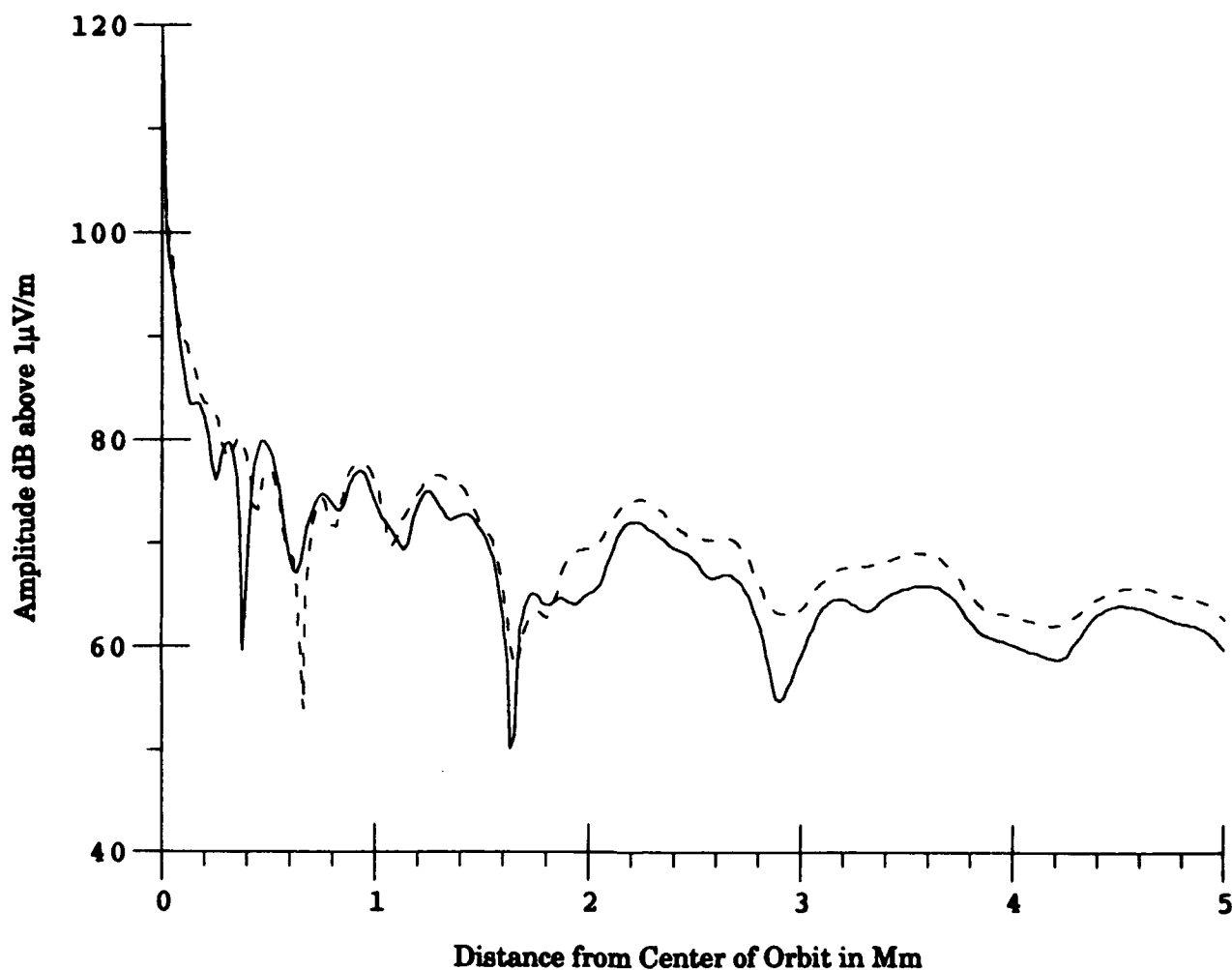


Figure 9. Vertical field strength  $E_z$  at sea level as a function of distance associated with  $\Delta V = 30\%$ . The solid and the dashed curves correspond to  $\psi = 90^\circ$  and  $\psi = 270^\circ$ , respectively. Nocturnal easterly 22 kHz propagation from a point dipole approximation of the dual trailing wire antenna.



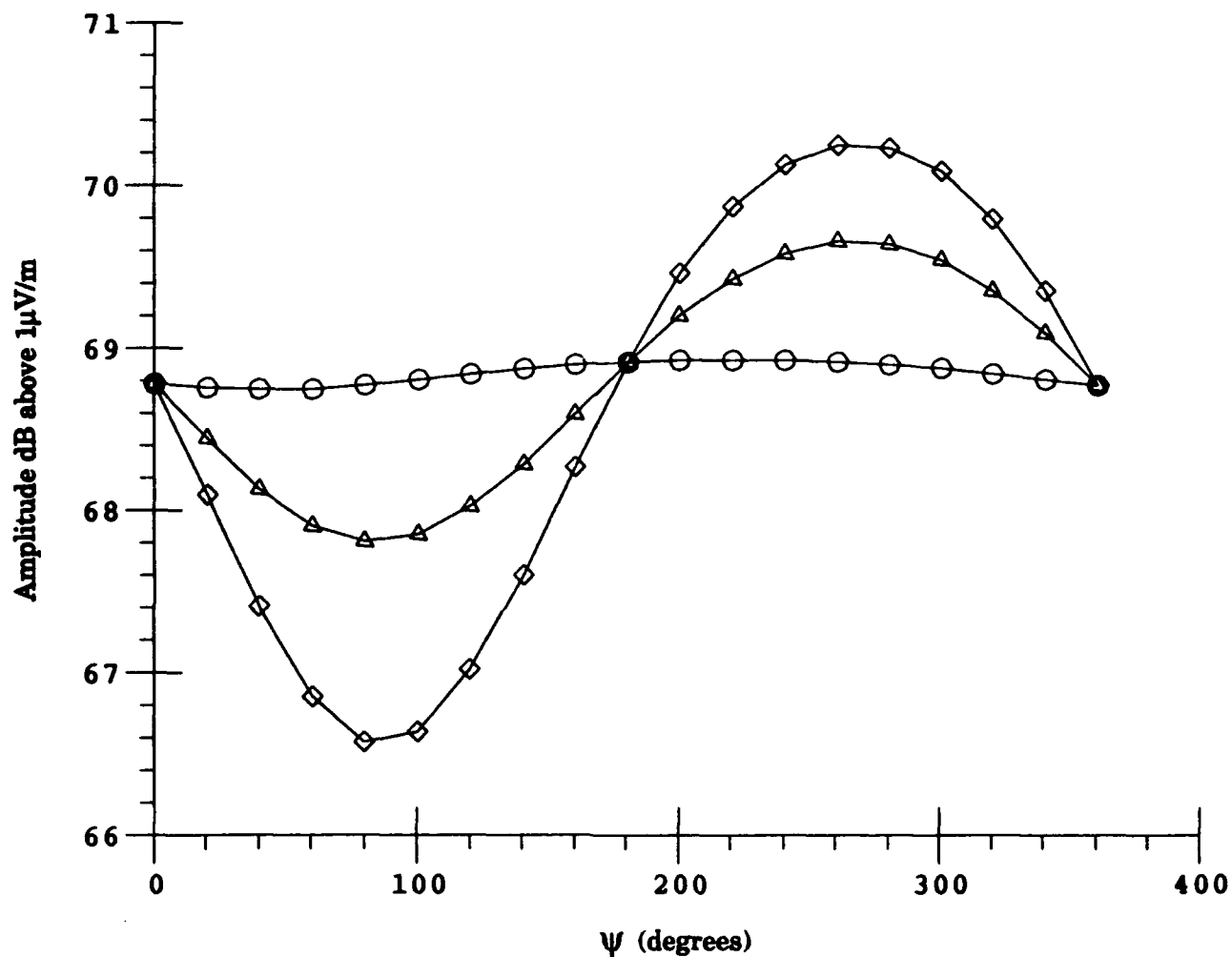


Figure 10. Yo-yo dependence of the vertical field strength  $E_v$  at 2.6 Mm. The circles correspond to  $\Delta V = 0\%$ , the triangles to  $\Delta V = 15\%$  and the diamonds to  $\Delta V = 30\%$ . Nocturnal easterly 22 kHz propagation from a point dipole approximation of the dual trailing wire antenna.

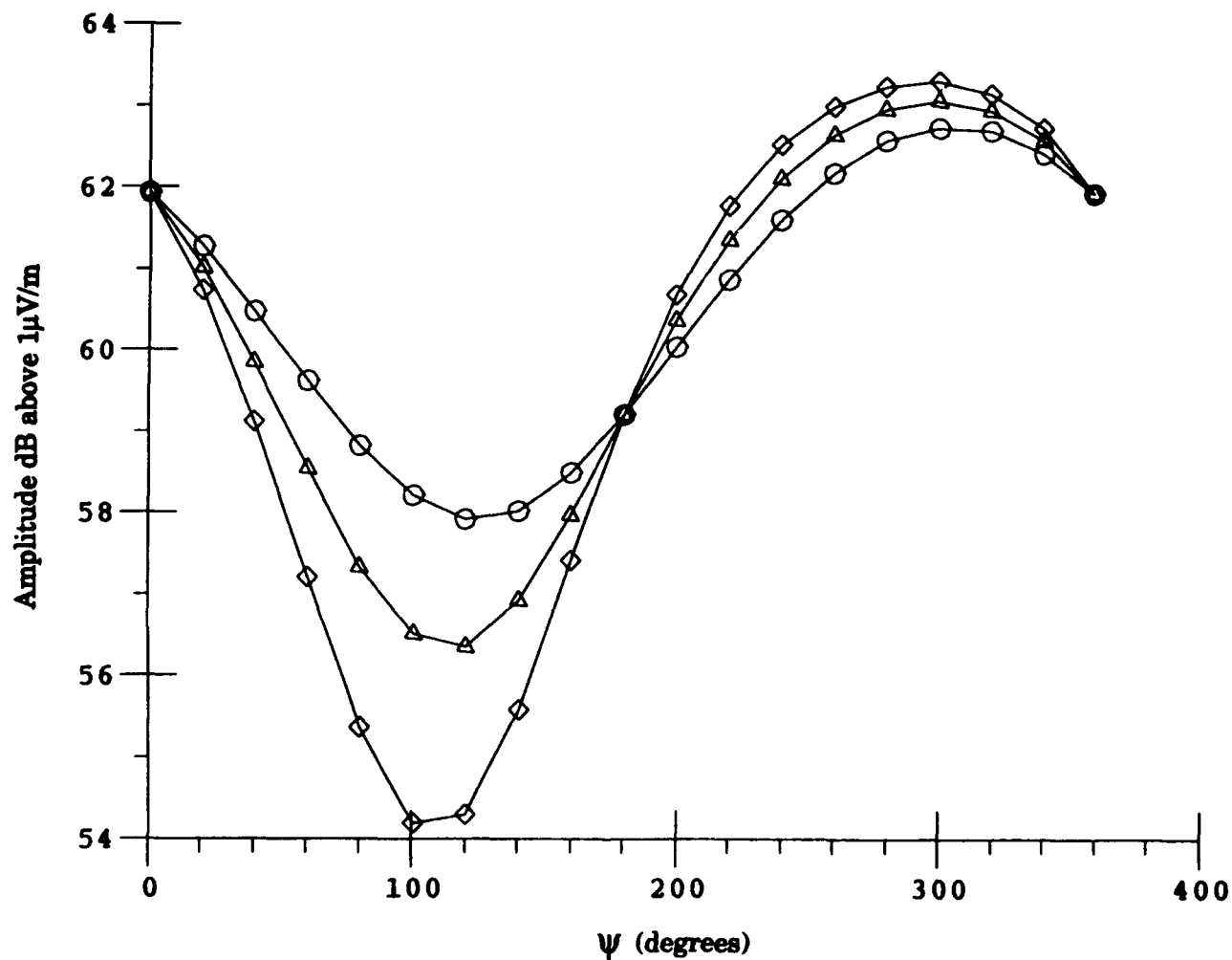


Figure 11. Yo-yo dependence of the vertical field strength  $E_z$  at 2.9 Mm. The circles correspond to  $\Delta V = 0\%$ , the triangles to  $\Delta V = 15\%$  and the diamonds to  $\Delta V = 30\%$ . Nocturnal easterly 22 kHz propagation from a point dipole approximation of the dual trailing wire antenna.

# **AN INVESTIGATION OF WIRE GRID AND SURFACE PATCH MODELING USING THE NUMERICAL ELECTROMAGNETICS CODE (NEC)**

**James K. Breakall**

Electrical and Computer Engineering Department  
The Pennsylvania State University  
University Park, PA 16802

**Richard W. Adler and Panos D. Elliniadis**

Electrical and Computer Engineering Department  
Naval Postgraduate School  
Monterey, CA 93943

## **ABSTRACT**

The Numerical Electromagnetics Code (NEC) was used to evaluate the admittance and the electric near and far fields of a monopole antenna mounted on a cubical box over a perfectly conducting ground plane. Two models of the box, employing surface patches and wire grids, were evaluated. The monopole was positioned at the center, the edge, and at a corner of the box's top surface. NEC admittance results were obtained and good agreement was found with experimental data and with results from PATCH, another independent electromagnetic modeling code. Results are presented in contour and 3-D formats for the near fields and polar format for the far field radiation patterns using surface patch and wire grid models in NEC. Excellent agreement was obtained for both approaches in NEC after finding the optimum number of patches and wire grid segmentation to obtain convergence. This paper provides guidelines for convergence for both modeling approaches and indicates a six-fold savings in run-time for the surface patch method. Furthermore, results are presented in modern graphical format for near field comparisons of the two NEC techniques.

## **I. INTRODUCTION**

The Method of Moments technique is the theoretical basis for the Numerical Electromagnetics Code (NEC), which is a code for the simulation and analysis of the electromagnetic response of antennas and other metallic structures [1]. NEC is the computer simulation tool that was used in this investigation of near fields.

Experimental and computational investigations were previously performed to determine the admittance characteristics of a monopole antenna mounted on a cubical conducting box of 0.1 m sides ( $\lambda/3$  at a frequency of 1 GHz) over a ground plane [2,3]. This simple geometrical model was used to simulate the basic shipboard topside environment of a ship's superstructure. The antenna, a 6 cm monopole ( $\lambda/5$  at the same frequency of 1

GHz), was tested for three different mounting positions on the top surface. Experimental data and numerically calculated results using the PATCH computer code of admittance for the 6 cm monopole antenna were presented versus frequency. PATCH is a recently developed frequency domain electromagnetic analysis code based on a Method of Moments solution to the Electric Field Integral Equation (EFIE) [4]. In this code objects are modeled by planar triangular patches which easily conform to surfaces and boundaries of general shape and allow variable patch densities over the surface of the object. This code can model open as well as closed surfaces which is a major advantage over previous Magnetic Field Integral Equation (MFIE) patch codes which only could model closed surfaces.

In this paper, the Numerical Electromagnetics Code (NEC) is used to evaluate the admittance and also the electric near and far field structure of the 6 cm monopole antenna mounted on the cubical box.

Convergence results are obtained and presented offering possible guidelines for more complex models. Also, this paper will present results of near fields for both surface patches and wire grid models in NEC using modern graphical formats. Additionally, it has been found in this paper that wire grid models take as much as six times the run-time of like surface patch models in NEC. It is hoped that all of these findings can form the basis of useful guidelines for further modeling of complex objects using NEC.

## II. BACKGROUND

The fields around an antenna may be divided into two regions, one near the antenna called the near field or Fresnel zone and one at a large distance called the far field or Fraunhofer zone [5]. The usually specified boundary between the near field and far field is the distance,  $r=2D^2/\lambda$  where  $D$  is the maximum length of the antenna in meters and  $\lambda$  is the wavelength in meters. The distance from the surface of the antenna to this boundary is called the near field region, while beyond this boundary the region is called the far field. The near field region can be further divided into two subregions, the reactive and radiating near field. The reactive near field usually extends to  $\lambda/2\pi$  from the antenna's surface, while in practice a distance of  $\lambda$  is used to represent this boundary. The phase of the magnetic and electric field is almost in quadrature in regions within a wavelength of the antenna (reactive near field). Beyond the distance of a wavelength, the electric and magnetic fields are propagating in phase (radiating near field) until the far field is reached. In the far field, the shape of the field pattern is independent of the distance, while in the near field the shape depends on this distance.

A description of the numerical codes used in this and the previous work follows. PATCH and NEC are both method of moments computer codes based on either/or both the EFIE or MFIE solutions of the full boundary solution of Maxwell's equations for current density on either cylindrical conductors (wires) or infinitesimally thin flat plates (patches).

PATCH, used in the work reported previously, will calculate both electromagnetic scattering and radiation from objects of arbitrary shape using the EFIE method. This allows modeling objects that are either open or closed and uses planar triangular patches conforming to the surface of the body. The numerical implementation in the code uses subdomain basis expansion functions placed on adjacent pairs of the triangular patches in the Method of Moments procedure. Details concerning this formulation can be found in [6].

NEC, will calculate both electromagnetic scattering and radiation for thin-wire structures of small cylindrical volume using the EFIE method or large closed voluminous and smooth bodies using the MFIE method. For thin structures such as plates or objects which have an opening, the EFIE method provides reasonable accuracy with wire grids having adequate spacing density. A coupled hybrid approach of both EFIE and MFIE is used to model structures containing both wires and closed surfaces and allows a connection of the wires to the surface. Details concerning the derivation of these methods can be found in [7]. Other details involving the choice of the basis functions, current and charge conditions, and capabilities used in NEC can be found in [8,9,10,11,12].

NEC and PATCH both will give solutions for current distributions as mentioned and therefore impedance and admittance at the feedpoint of a voltage excitation. This paper compares surface patch results from NEC with those from PATCH and measurements on values of admittance versus frequency. Wire grid modeling comparisons for admittance using NEC have been performed previously [13]. Furthermore, this work here computes near and far field results which are produced with both wire grid and surface patch modeling in NEC and it is hoped that they can be used to further validate PATCH and other codes.

### III. RESULTS

The optimum model for complex structures can be estimated by varying the segment and patch density and observing the results and the convergence of the solution [14]. In the case of an edge-mounted antenna, the accuracy of the results is expected to depend upon the size of the segments and patches near the edge. Smaller segments and patches are suggested at edge areas since the current magnitude may vary rapidly in this region.

The numerical model of a cubical five-sided box of 0.1 meters per side was constructed using NEC (the bottom was not included as a surface since the box was placed on a perfectly conducting ground plane). A 6 cm monopole antenna was placed at the center, at the edge (3.63 cm from center) and at a corner (5.14 cm on a diagonal from center), as shown in Figures 1a and 1b for wire grids and patches respectively.

The first part of the investigation checked the input impedance using NEC as patch density was varied. The number of patches on the top of the box was varied in search of an optimum value of surface samples, which would later be used for near field calculations.

The top was subdivided to retain symmetry, as much as possible, and to closely match positions of the antenna on the experimental model.

#### **A. Monopole at the Center**

The monopole was divided into five segments and placed at the center of the top surface. The top surface was divided into 25 (5x5), 49 (7x7), 81 (9x9), and 121 (11x11) patches. Varying the subdivision of the top surface in this manner provided convergence in the results which can easily be seen in Figures 2a and 2b for conductance and susceptance, respectively. Since the current density will change most rapidly at the connection of the monopole, the subdivision of the connection patch is automatically divided by four. It was found that convergence was obtained with the 9x9 subdivision and the correlation of NEC and PATCH results with the measurements is quite good as shown in Figure 2c.

#### **B. Monopole at the Edge**

The monopole was attached to an edge at a distance 3.63 cm from the center which corresponds closely to the actual configuration. The difference in distance for the position of the monopole in the NEC model compared to the actual physical geometry is 0.13 cm. A subdivision of the top surface into 81 (9x9) patches produced well-converged results. In Figure 3a are shown the NEC results as well as measurements and PATCH results. It can be seen that good agreement is obtained with measurements and PATCH's conductance values. NEC and PATCH have almost identical performance for susceptance as compared to measurements.

#### **C. Monopole at Corner**

The 6 cm monopole in the NEC model was placed at the corner, 5.14 cm on the diagonal from the center and fed at the base. The position of the monopole for the experimental model was 5.15 cm. The top surface was divided into 121 (11x11) patches to obtain well-converged results.

NEC and PATCH results compared to measurements are presented in Figure 3b. NEC is in excellent agreement with PATCH and measurements in both conductance and susceptance, and in the range of 1.15 to 1.40 GHz, both codes are virtually identical.

#### **D. Near Electric Field**

Near fields are more difficult to calculate in NEC than far fields. When calculating radiation in close proximity to an antenna, terms in the field expressions with powers of  $1/r^2$  ( $r$  is the distance from the origin of the antenna to the field point) are appreciable in magnitude compared to the  $1/r$  dependent terms which are dominant in the far field. The near field is thus very dependent on the charge density and the current while the far field is mainly dependent on the current.

For near field calculations, NEC computes the magnitude and phase of each component,  $E_x$ ,  $E_y$ , and  $E_z$  separately and a modification was made to also calculate the magnitude of the peak electric field (E-Total) in (V/m), which is the vector sum of the three components  $E_x$ ,  $E_y$ , and  $E_z$ . Using the optimum models in NEC for the monopole at the center (9x9 patches), at the edge (9x9 patches), and the corner (11x11 patches), the near field was investigated.

In order to compare NEC near fields for the monopole on the box with known theoretical understanding, we consider a linear current element  $I = I_0 e^{j\omega t}$  of length  $z$  oriented in the  $z$  direction and with amplitude  $I_0$  located at the origin as in Figure 4. This antenna is a known simple radiating structure but it will demonstrate basic properties of the near electric field for all small linear antennas. The complete electric field intensity of the antenna is:

$$\begin{aligned} \bar{E} = & \frac{-I_0 \Delta z}{2\pi} j \frac{\sqrt{\mu_0}}{k_0 \sqrt{\epsilon_0}} \cos \theta \left( \frac{jk_0}{r^2} + \frac{1}{r^3} \right) e^{-jk_0 r} \bar{i}_r + \\ & \frac{-I_0 \Delta z}{4\pi} j \frac{\sqrt{\mu_0}}{k_0 \sqrt{\epsilon_0}} \sin \theta \left( \frac{-k_0^2}{r} + \frac{jk_0}{r^2} + \frac{1}{r^3} \right) e^{-jk_0 r} \bar{i}_\theta \end{aligned} \quad (1)$$

The only part of the field dominant in the expression for the far field radiated power is that part consisting of the terms varying as  $r^{-1}$ , that is

$$\bar{E} = \frac{jk_0 I_0 \Delta z \sqrt{\mu_0}}{4\pi r \sqrt{\epsilon_0}} \sin \theta e^{-jk_0 r} \bar{i}_\theta \quad (2)$$

The parts of the field varying as  $r^{-2}$  and  $r^{-3}$  are important in the near field. Consequently the terms that are functions of distance  $r$  of the E-Field in the above equations are:

$$\left( -\frac{k_0^2}{r} + j \frac{k_0}{r^2} + \frac{1}{r^3} \right) \quad \text{along the x or y-axis} \quad (3a)$$

$$\left( \frac{jk_0}{r^2} + \frac{1}{r^3} \right) \quad \text{along the z-axis.} \quad (3b)$$

All the other terms are phase terms or constants. Generally, the magnitude of the electric field has an  $r$ -dependence which can be expressed as:

$$|E(r)| = c \sqrt{\left(\frac{1}{r^3} - \frac{k_0^2}{r}\right)^2 + \left(\frac{k_0}{r^2}\right)^2} \quad \text{along the x or y-axis} \quad (4a)$$

$$|E(r)| = c \sqrt{\left(\frac{1}{r^3}\right)^2 + \left(\frac{k_0}{r^2}\right)^2} \quad \text{along the z-axis} \quad (4b)$$

where  $c$  is a proportionality constant used to normalize the electric field to the starting position.

FORTTRAN programs were developed to plot the magnitude and phase contours of the near electric field using NEC output data [14]. For reference, two simple antennas with analytical results are examined: (1) a 0.15 m ( $\lambda/2$ ) dipole antenna in free space, and (2) a 0.06 m ( $\lambda/5$ ) monopole antenna over a perfect ground plane which corresponds to the same monopole mounted on the cubical box. The near electric field contours are displayed in a section of a plane in 3-dimensional space. The results are shown in Figures 5a and 5b. It is seen that the shape of near electric field is the same for the dipole and monopole antennas. The ratio of the maximum current for the dipole to the monopole as computed by NEC with both fed from a 1 Volt excitation source is -7.0 dB. Therefore, absolute field values for the dipole should be increased by 7.0 dB to make comparisons to the monopole fields with both having the same current. The phase plots of the electric fields for the  $\lambda/2$  dipole and 0.06 m monopole are shown in Figures 6a and 6b and show that both antennas have a smooth spherical wavefront pattern.

NEC solutions of the near electric field of the monopole on the box center are shown in Figures 7a and 7b for the magnitude of the total electric field and the phase of the  $E_z$  component. The formation of maxima and nulls can also be observed. A maximum occurs at  $60^\circ$  in elevation from the box surface while a null is seen at about  $30^\circ$ . The main lobe starts to develop at a distance  $1\lambda$  (0.3m) from the antenna. Beyond this point, the main lobe has the same shape independent of distance. Within a distance of  $2\lambda$  the pattern shape has not yet fully developed.

In order to gain insight into the electric near field variations and where the maxima and minima occur, a 3-dimensional plot is presented in Figure 8. The plot displays a surface whose elevation points represents field strength in the upper portion, and a normal 2-D contour plot "projection" in the lower portion. In this figure the vertex that corresponds to the pointed "spike-like" area of the surface is the origin where the antenna is mounted. The decay of the field as the observation point moves away from the monopole source is expected. The null "trough" can easily be seen in this type of representation.



Figures 9a and 9b show results for the edge-mounted monopole in the same format as the center-mounted case. The following comments will amplify differences from the center case and interesting features of the field distributions. The edge-mounted geometry provides a larger planar surface (box top) in front of the monopole view plane and results in the following differences with respect to the center-mounted geometry:

1. The z-axis peak field is somewhat greater at given distances from the box surface. This is caused by a larger  $E_x$  component.
2. The elevation plane null is not as deep.
3. The phase contours for the z component show evidence of two close-in nulls, but very small "phase wrinkles" beyond the near zone.
4. The contour plot depicts a more uniform field overall, with a less severe null.

The corner-mounted monopole near field plots are in Figures 9c and 9d. The fields for this case fall in between the center and edge-mounted field configurations. That is:

1. The elevation plane null is between the nulls of the other two configurations.
2. "Phase wrinkles" show one null, as does the center-mounted case, but it is not as severe.
3. Contour plots indicate a wider field pattern with a fairly uniform distribution away from the monopole.

#### **E. Far Field**

Far field radiation patterns were calculated and are presented in Figures 10a and 10b (Monopole at center), 10c and 10d, (Monopole at edge), and 10e and 10f, (Monopole at corner). In Figure 10a, the vertical pattern of the monopole at the center shows that the maximum gain is very close to 5.15 dBi, the theoretical value for a monopole over an infinite perfectly conducting ground plane. In Figure 10b, the horizontal pattern shows omnidirectional radiation from the electrically small box-monopole configuration which is not expected to contribute much directionality in azimuth.

The results of vertical and horizontal patterns for the edge-mounted monopole (Figures 10c and 10d) and the corner-mounted monopole (Figures 10e and 10f) display unsymmetrical patterns.

## F. Wire Grid Modeling

Solid surfaces can be modeled in NEC with a grid of wires, with the restriction that the grid cells are to be small in terms of a wavelength. Wire grid modeling guidelines are given in [1,15,16]. For the wire grid modeling technique, typical run-times of the box-monopole configuration have been found to take up to six times those of the surface patch models. The box of Figure 1 is modeled as a five-sided wire grid box of 0.1 m ( $\lambda/3$  at 1 GHz) per side having cells of 0.0125 by 0.0125 meters. The 0.06 m monopole antenna was divided into 5 segments and placed on top of the wire grid box at the center, edge (3.75 cm from center) and corner (5.3 cm on the diagonal). The antenna was fed at the base segment for all cases. The wire grid geometry of [13] was used for calculations of the near electric field in the present study. This geometry produced good results for admittance compared with experimental data [2,3]. Magnitude and phase contour plots of near electric field for the wire grid box case with the same field point locations used in the surface patch model are shown for the monopole mounted at the center. The excellent agreement of near fields (Figures 11a and 11b) for the wire grid box with the center-mounted monopole to those of the surface patch model (Figures 7a and 7b) attests to the equivalence of the two numerical models. Differences are less than 1 dB, a value which is difficult to measure. The other wire grid cases for the monopole at the edge and corner also were compared to the surface patch models and excellent agreement was likewise obtained with differences of 1 to 1.5 dB [14]. The differences are expected to be attributed to a more accurate rendition of edge effects for the wire grid model versus the patch model.

## IV. CONCLUSIONS

The goal of this investigation was to accurately predict admittance, near and far fields using the Numerical Electromagnetics Code (NEC).

Since few validation benchmark results for near fields were available, an exercise was undertaken using NEC in order to determine optimum models. Box-like structures were analyzed using surface patch and wire grid modeling techniques. The simulation models consisted of a  $\lambda/5$  monopole mounted on the top of a  $\lambda/3$  box at three different locations: center, edge, and corner. Optimum models were selected by varying the patch density on the top surface of the box and observing the convergence of the solution and comparison with measurements and PATCH results for input admittance, in order to ensure the validity of the models and improve the confidence in near field predictions. Optimum NEC models for the three different mounting geometries were found to be:

- CENTER and EDGE:  $9 \times 9 = 81$  patches on top ( $0.0013 \lambda^2$  patch area)
- CORNER:  $11 \times 11 = 121$  patches on top ( $0.0009 \lambda^2$  patch area)

Even though edges are not modeled in the surface patch technique [1], this study proves that, for positions very close to an edge, good results can be obtained by careful subdividing (no special subdivision of smaller patches in the vicinity of the edge or corner was required).

Algorithms were developed to produce near electric field (magnitude and phase) contours and 3-D plots [14]. The near field for the monopole on the box has similar characteristics in magnitude and phase as the monopole over a ground plane except in the region where nulls occur from box radiation and diffraction effects. The edge/corner-mounted geometries produced slightly different near field contours compared to the center-mounted geometry. Surface patch and wire grid models for NEC gave essentially similar results for near fields.

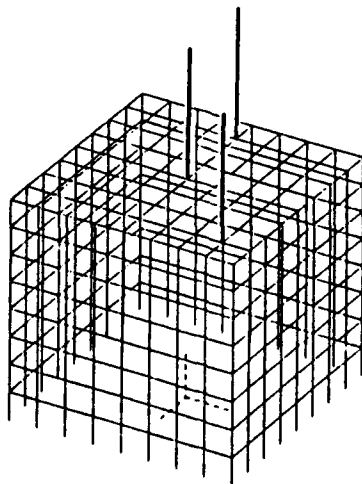
Previously, generalized guidelines for near field modeling had not been developed for NEC and the use of wire grid and surface patch modeling for near field determination was approached with caution. Guidelines developed in this study, as well as the results of the near field behavior of the monopole antenna on the conducting box, can be used for future investigations on more complex structures.

The present study is an important step in the direction of modeling the effects of the near field of antenna structures.

## REFERENCES

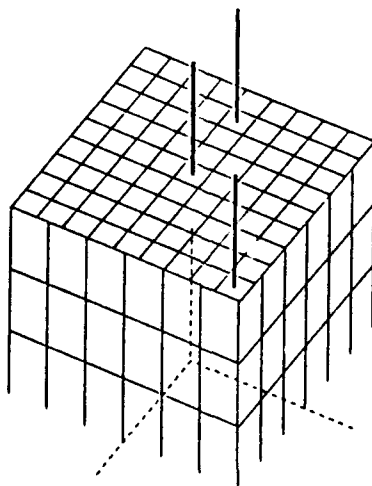
- [1] G. J. Burke and A. J. Poggio, "Numerical electromagnetics code (NEC) - Method of Moments," Naval Ocean Systems Center, Tech. Document 116, January 1980.
- [2] S. Bhattacharya, "A study of the admittance characteristics of a monopole antenna attached to a conducting box," Master's Thesis, University of Houston, pp. 11-13, pp. 71-73 and pp. 86-88, May 1986.
- [3] S. Bhattacharya, S. A. Long, and D. R. Wilton, "The input impedance of a monopole antenna mounted on a cubical conducting box," *IEEE Trans. Antennas Propagat.*, vol. AP-35, No. 7, pp. 756-761, July 1987.
- [4] W. A. Johnson, D. R. Wilton, and R. M. Sharpe, "Patch code user's manual," Sandia National Laboratories, Rep. SAND87-2991, May 1988.
- [5] A. D. Yaghjian, "An overview of near-field antenna measurements," *IEEE Trans. Antennas Propagat.*, vol. AP-34, No. 1, pp. 30-43, January 1986.
- [6] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propagat.*, vol. AP-30, No. 3, pp. 409-418, May 1982.
- [7] A. J. Poggio and E. K. Miller, "Integral equation solutions of three-dimensional scattering problems," Chapt. IV in *Computer Techniques for Electromagnetics*, Edited by R. Mittra, Pergamon Press, NY, 1973.
- [8] J. K. Breakall, G. J. Burke, and E. K. Miller, "The Numerical Electromagnetics Code (NEC)," EMC Symposium & Exhibition, Zurich, Switzerland, March 1985.
- [9] Y. S. Yeh and K. K. Mei, "Theory of conical equiangular spiral antennas," Part I - Numerical Techniques, *IEEE Trans. Antennas Propagat.*, vol. AP-15, No. 5, p. 634, September 1967.
- [10] A. R. Neureuther et al., "A comparison of numerical methods for thin wire antennas," presented at the 1968 Fall URSI Meeting, Dept. of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1968.
- [11] E. K. Miller, R. M. Bevensee, A. J. Poggio, R. Adams, and F. J. Deadrick, "An evaluation of computer programs using integral equations for the electromagnetic analysis of thin wire structures," Lawrence Livermore Nat. Lab. Rep. UCRL-75566, March 1974.

- [12] N. C. Albertsen, J. E. Hansen, and N. E. Jencen, "Computation of spacecraft antenna radiation patterns," The Technical University of Denmark, June 1972.
- [13] C. R. Molina, "Numerical electromagnetic models of cube-shaped boxes - An investigation for near field prediction of HF shipboard environments," Master's Thesis, Naval Postgraduate School, Monterey, California, December 1987.
- [14] P. D. Elliniadis, "An investigation of near fields for HF shipboard antennas - Surface patch and wire grid modeling using the Numerical Electromagnetics Code," Master's Thesis, Naval Postgraduate School, Monterey, California, December 1988.
- [15] G. J. Burke, "Enhancements and limitations of the code NEC for modeling electrically small antennas," Lawrence Livermore Nat. Lab. Rep. UCID-20970, January 1987.
- [16] G. J. Burke, "Treatment of small wire loops in the Method of Moments code NEC," Lawrence Livermore Nat. Lab. Rep. UCID-21196, October 1987.



THETA = 60.00 PHI = 30.00 ETA = 90.00

**Figure 1a. Monopole at the Center, Edge, and Corner of the Wire Grid Box Model.**



THETA = 45.00 PHI = 30.00 ETA = 90.00

**Figure 1b. Monopole at the Center, Edge, and Corner of the Surface Patch Box Model.**

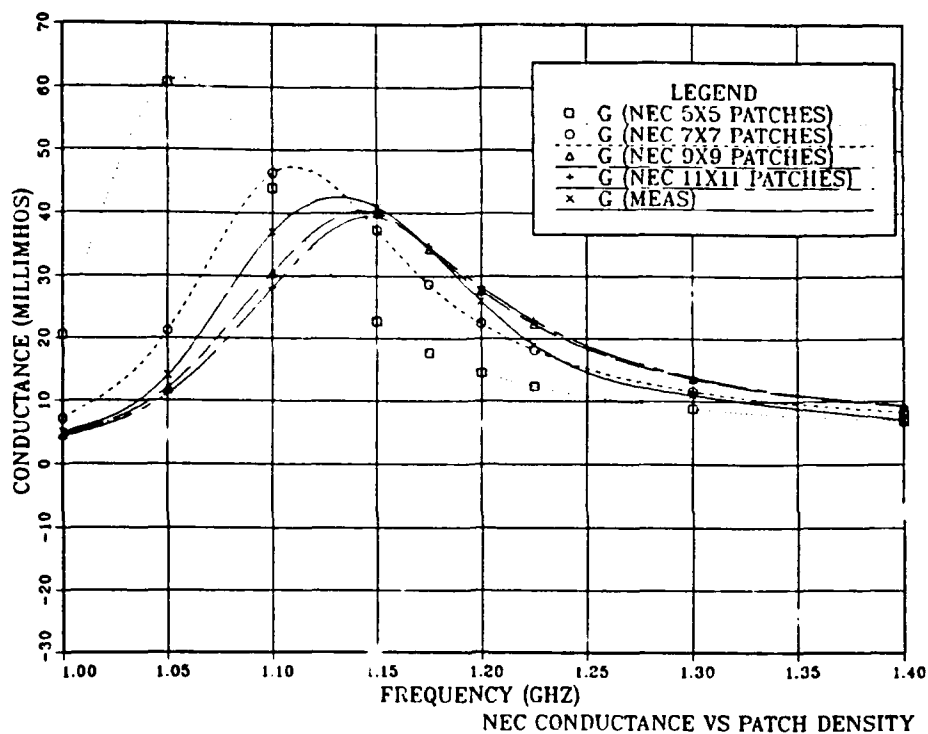


Figure 2a. Monopole at Center of Patch Box, NEC Conductance vs. Patch Density.

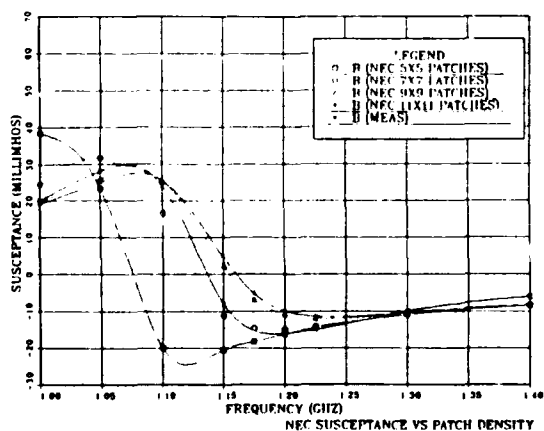


Figure 2b. Monopole at Center of Patch Box, NEC Susceptance vs. Patch Density.

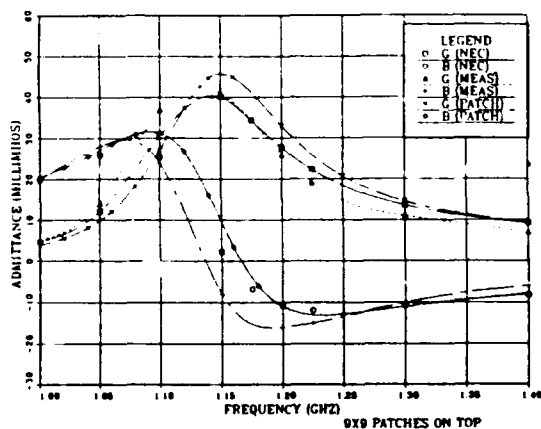


Figure 2c. Monopole at Center of Patch Box, NEC Admittance vs. Measurements and PATCH Code.

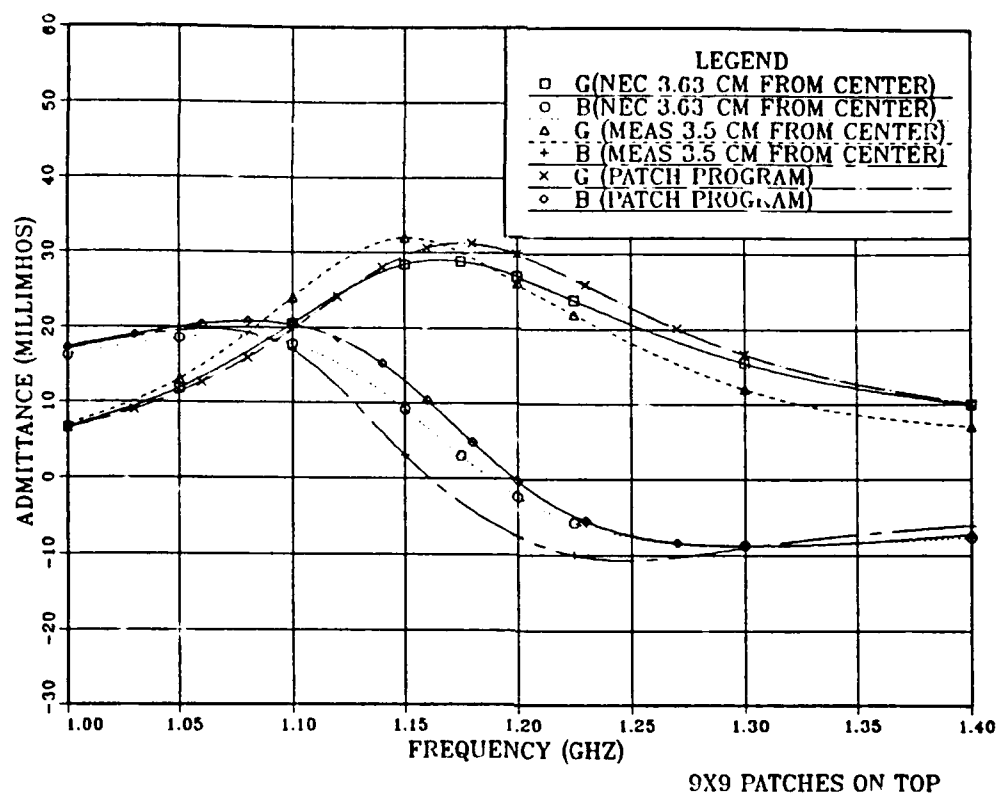


Figure 3a. Monopole at Edge of Patch Box, NEC Admittance vs. Measurements and PATCH Code.

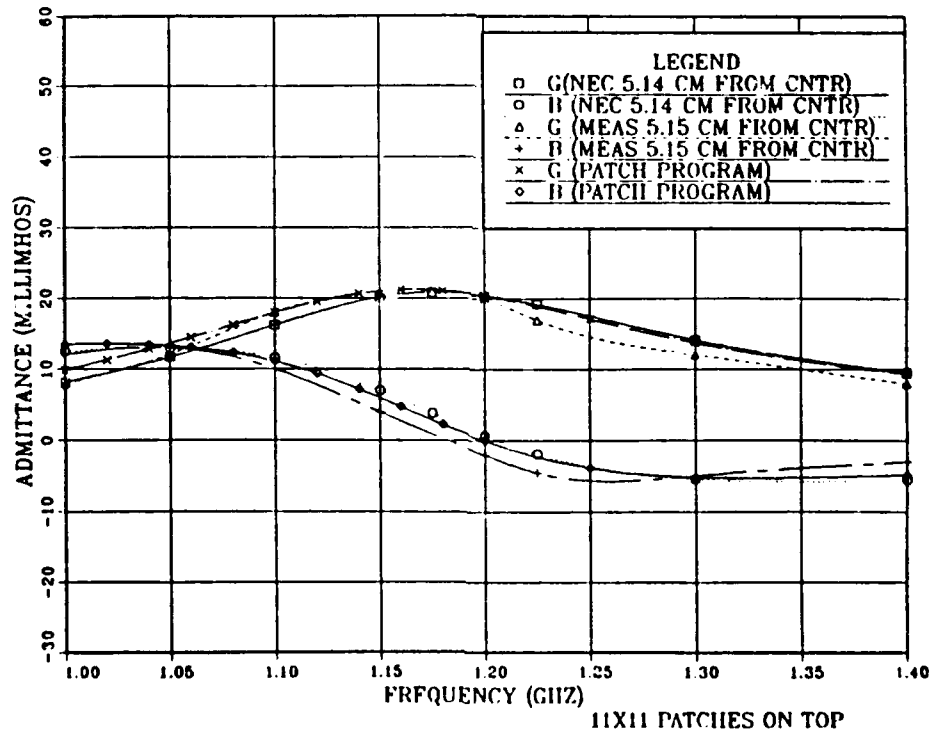


Figure 3b. Monopole at Corner of Patch Box, NEC Admittance vs. Measurements and PATCH Code.



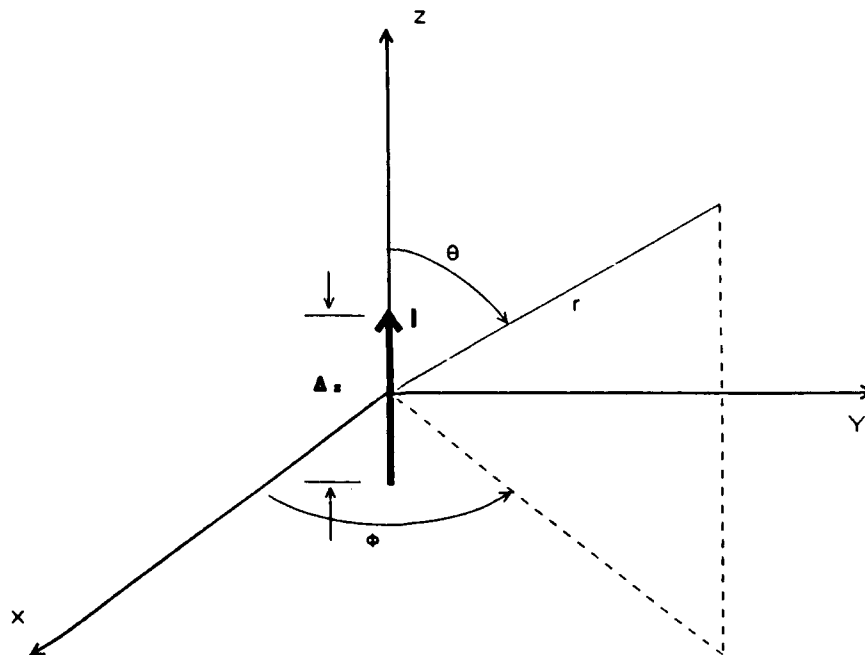


Figure 4. A Linear Current Radiator.

CONTOUR E-FIELD (DB REFER TO 1V/M)  
DIPOLE LENGTH  $\lambda/2$  ON Z AXIS  
FREE SPACE-FREQ-1GHZ

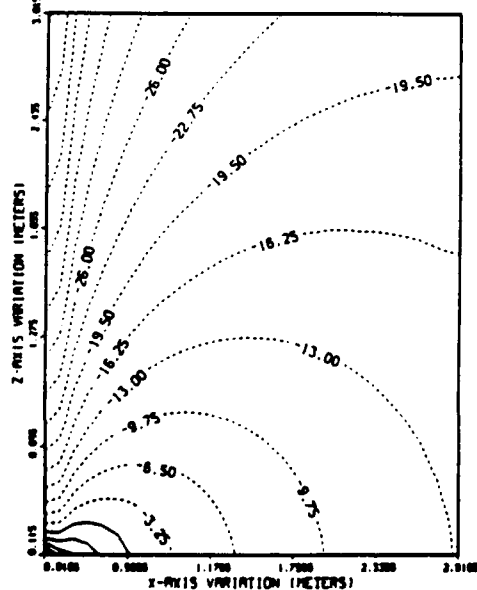


Figure 5a. Total E-Field Contours, Dipole  $\lambda/2$  on Z-Axis in Free Space.

CONTOUR E-FIELD (DB REFER TO 1V/M)  
MONOPOLE 6 CM ON Z AXIS  
OVER PERFECT GROUND-FREQ-1GHZ

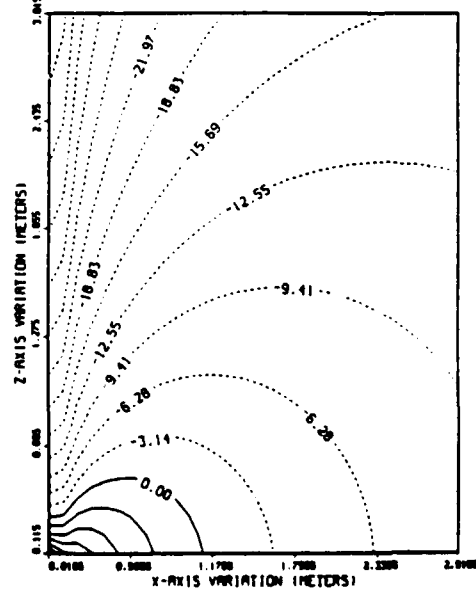


Figure 5b. Total E-Field Contours, Monopole 6 cm Over Perfect Ground.

COPY AVAILABLE TO DTIC DOES NOT PERMIT FULLY LEGIBLE REPRODUCTION

PHASE OF X COMPONENT OF E-FIELD  
DIPOLE  $\lambda/2$  ON Z AXIS  
IN FREE SPACE-FREQ-1GHZ

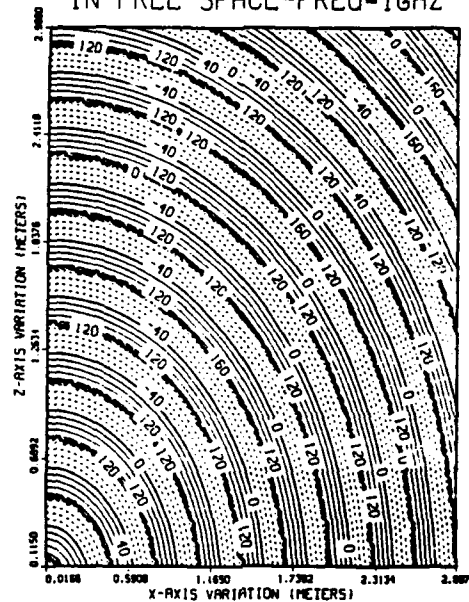


Figure 6a. Z-Component, E-Field Phase Contours, Dipole  $\lambda/2$  in Free Space.

PHASE OF Z COMPONENT OF E-FIELD  
MONOPOLE 6CM ON Z AXIS  
OVER PERFECT GROUND-FREQ-1GHZ

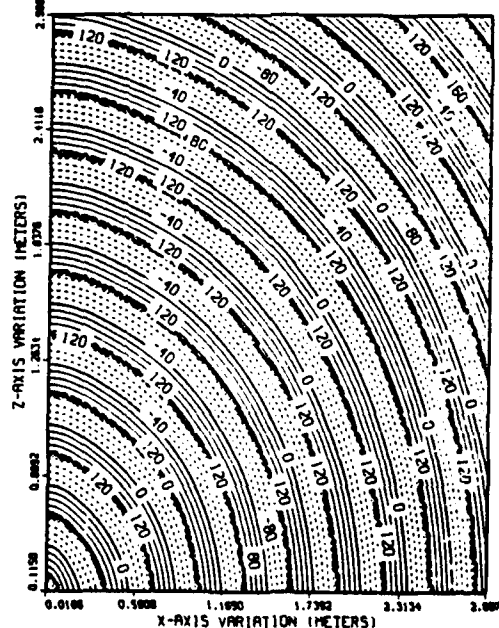


Figure 6b. Z-Component, E-Field Phase Contours, Monopole 6 cm Over Perfect Ground.

COPY AVAILABLE TO DTIC DOES NOT PERMIT FULLY LEGIBLE REPRODUCTION

CONTOUR E-FIELD (DB REF TO 1V/M)  
MONOPOLE 6 CM AT CENTER OF SPBOX

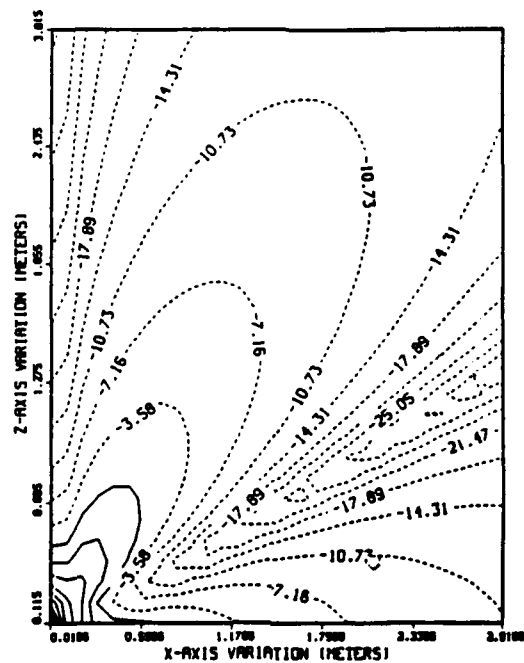


Figure 7a. Total E-Field Contours, Monopole at Patch Box Center.

PHASE OF Z COMPONENT OF E-FIELD  
MONOPOLE 6CM AT CENTER OF SPBOX

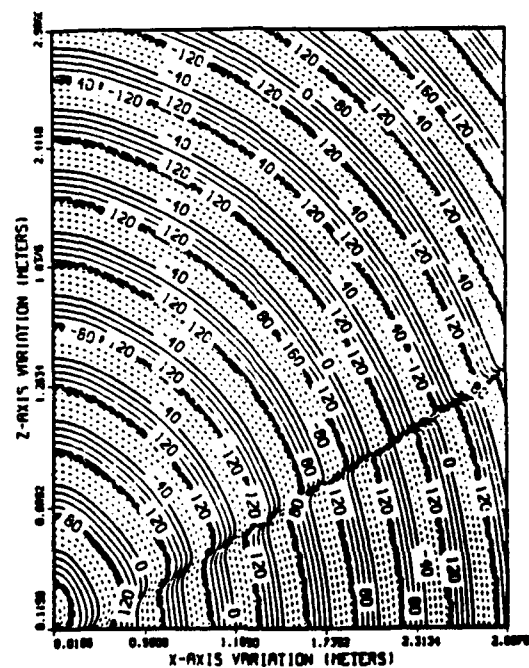


Figure 7b. Z-Component, E-Field Phase Contours, Monopole at Patch Box Center.

COPY AVAILABLE TO DTIC DOES NOT PERMIT FULLY LEGIBLE REPRODUCTION

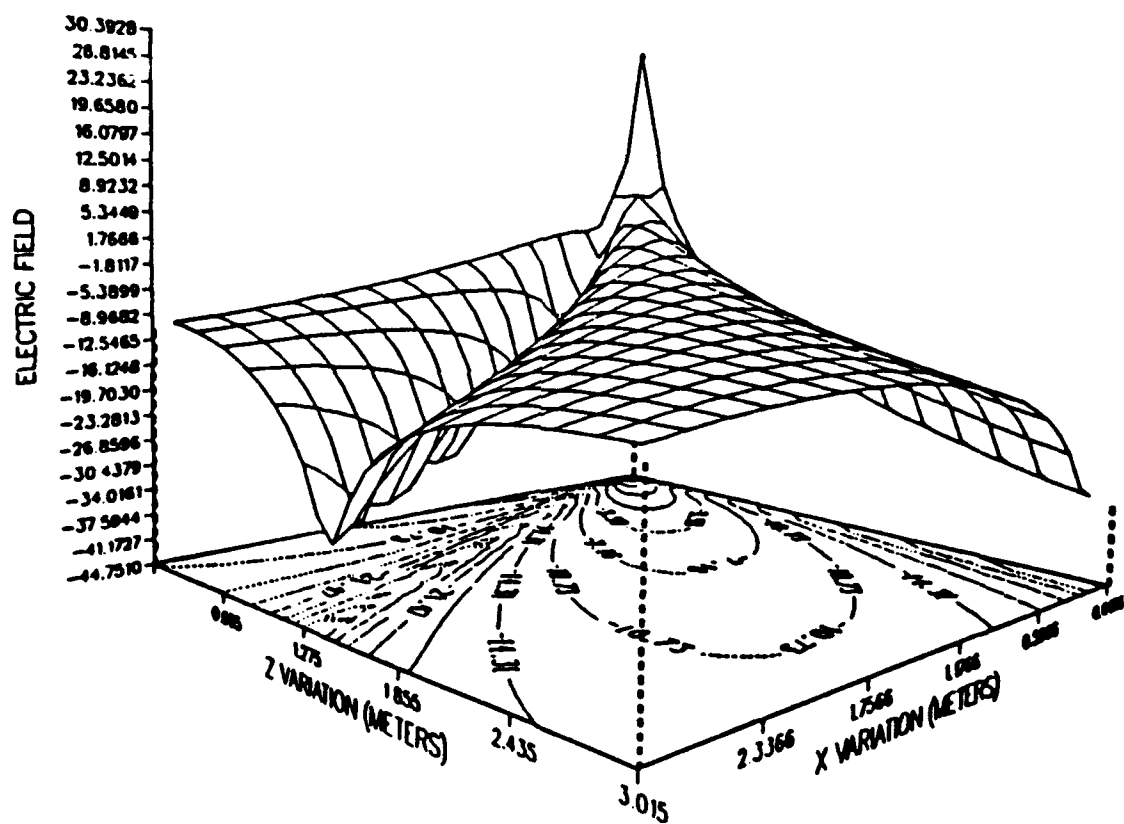


Figure 8. Total E-Field 3-D Plot, View Toward Monopole. Monopole at Patch Box Center.

CONTOUR E-FIELD (DB REF TO 1V/M)  
MONOPOLE 6CM AT EDGE (3.63 CM FROM CENTER) OF SPBOX

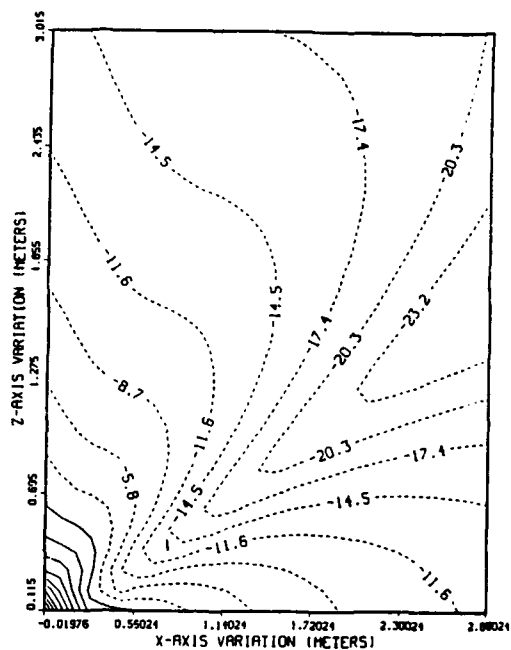


Figure 9a. Total E-Field Contours,  
Monopole at Patch Box Edge.

PHASE OF Z COMPONENT OF E-FIELD  
MONOPOLE 6CM AT EDGE OF SPBOX

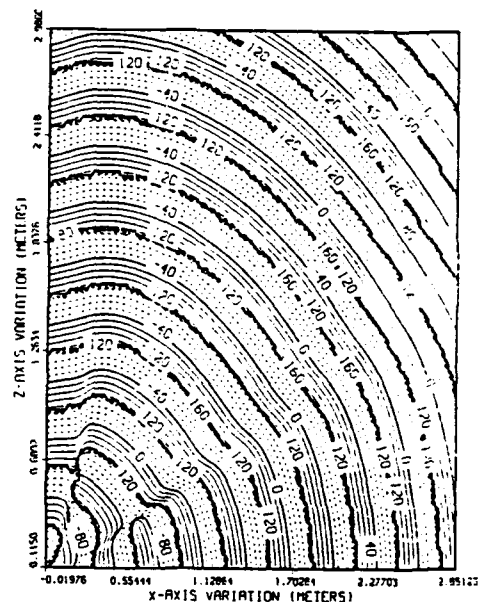
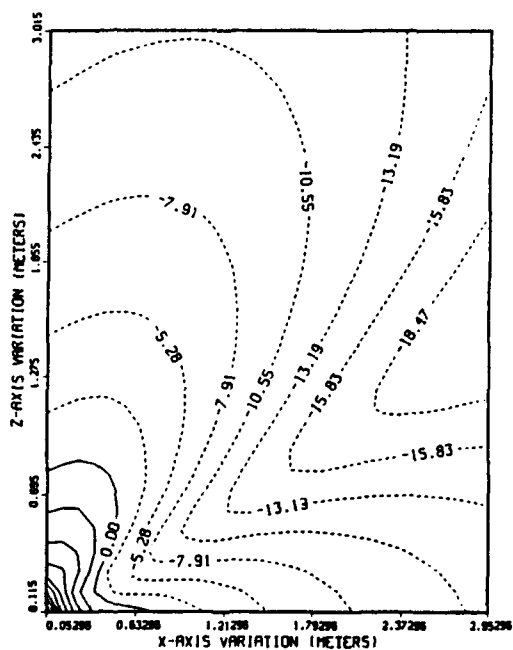


Figure 9b. Z-Component, E-Field Phase  
Contours, Monopole at Patch  
Box Edge.

CONTOUR E-FIELD (DB REF TO 1V/M)  
MONOPOLE 6CM AT CORNER (5.14CM ON DIAGONAL) OF SPBOX



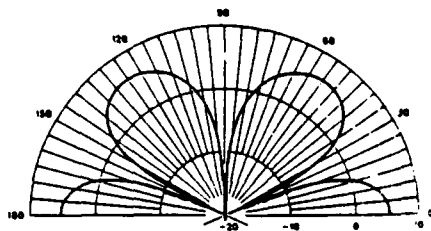


Figure 10a. Vertical Pattern, Monopole at Patch Box Center.

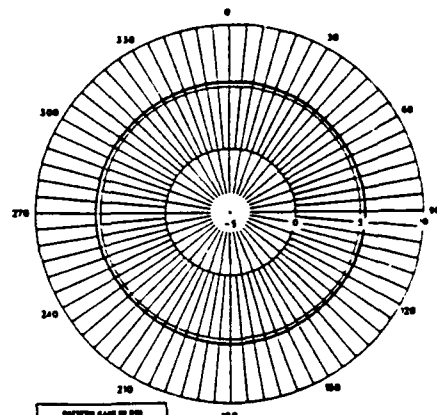


Figure 10b. Horizontal Pattern, Monopole at Patch Box Center.

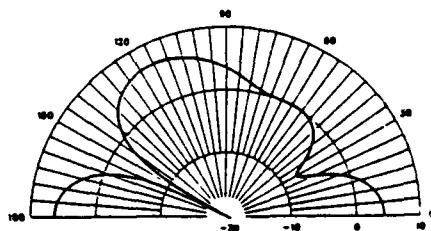


Figure 10c. Vertical Pattern, (X-Axis Cut), Monopole at Patch Box Edge.

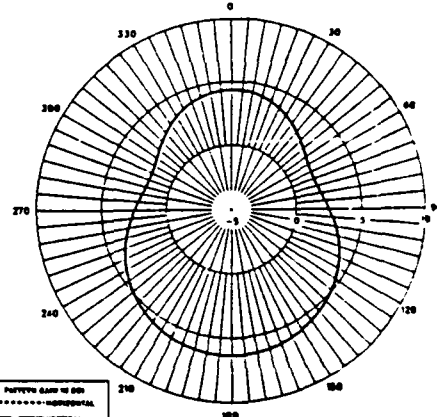


Figure 10d. Horizontal Pattern, Monopole at Patch Box Edge.

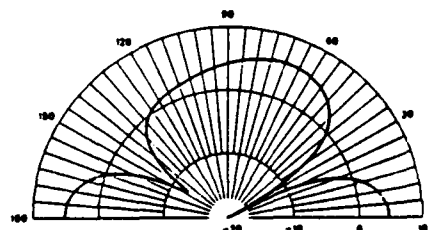


Figure 10e. Vertical pattern (45° Cut), Monopole at Patch Box Corner.

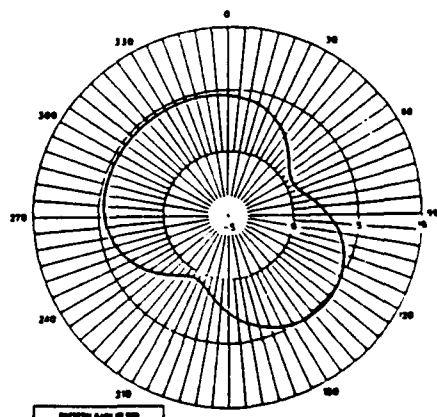


Figure 10f. Horizontal Pattern, Monopole at Patch Box Corner.

CONTOUR E-FIELD (DB REF TO 1V/M)  
MONOPOLE 6CM AT CENTER OF WIRE GRID BOX  
FREQ=1GHZ

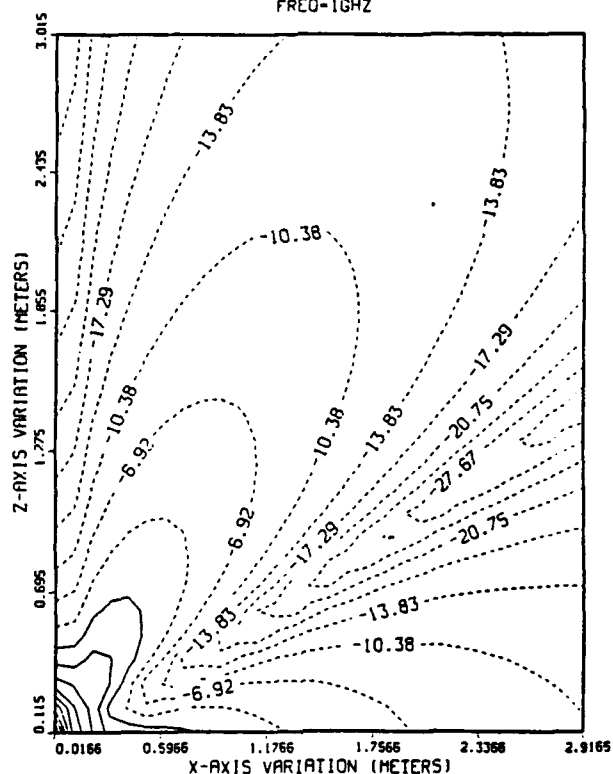


Figure 11a. Total E-Field Contours, Monopoles at Wire Grid Box Center.

PHASE OF Z COMPONENT OF E-FIELD  
MONOPOLE 6CM AT CENTER OF WIRE GRID BOX  
FREQ=1GHZ

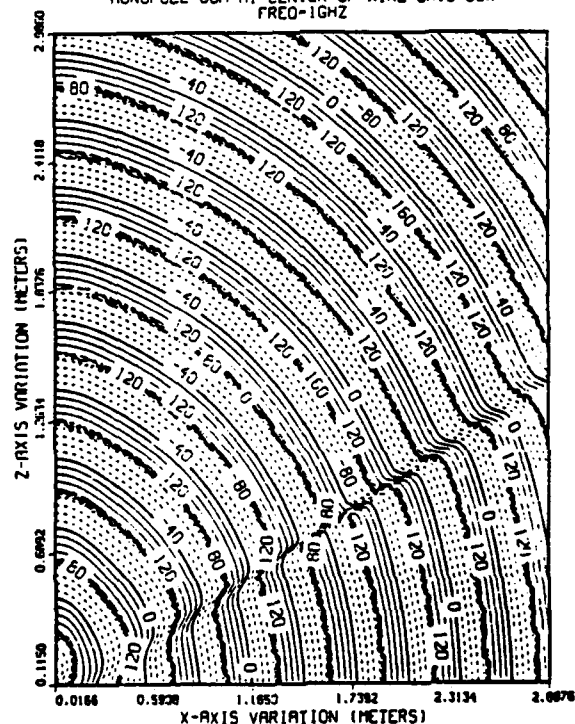


Figure 11b. Z-Component, E-Field Phase Contours, Monopole at Wire Grid Box Center.

**AN INTEGRATED ENVIRONMENT FOR THE NUMERICAL MODELING  
OF COMMUNICATION ANTENNAS BASED ON  
RELATIONAL DATABASES**

**Virginia Stover  
Department of Mathematics and Computer Science  
University of San Diego  
San Diego, CA**

**James C. Logan  
Research and Development Division  
Naval Command, Control and Ocean Surveillance Center  
San Diego, CA**

**Abstract**

As modeling systems mature, they become larger, more complex, and more difficult to maintain. Modeling tools increase in number and complexity. Frequently they are written in different languages and require data in different formats. Databases also increase in size as modeling systems are applied to new and more complex problems. Engineers spend large amounts of money trying to integrate tools and data that are basically incompatible. Unfortunately, budgets do not grow at the same rate as the complexity of our modeling systems and databases. To maintain productivity, it is necessary to design modeling environments that can handle large amounts of data in flexible ways and are simple to maintain and upgrade.

This paper describes a new environment developed by the authors for the modeling of communication antennas based on a relational database management system. This approach simplifies the task of integrating a set of heterogeneous programs with incompatible data formats. The relational database provides a common store for all modeling objects including the antenna, platform, ground, electromagnetic sources, currents, charges, and fields, and model history. The database management system provides the organization, storage, and retrieval functions and some of the data input, validation and display functions for the antenna models. The main advantages of this approach are its ability to grow as new tools and capabilities are added, its portability to other machines and operating systems, and the ability it provides engineers to easily share data among themselves and with other modeling applications.

---

This work was conducted for the Naval Ocean Systems Center as part of the Navy Summer Faculty Research Program, a cooperative program with the American Association for Engineering Education (ASEE).

**Introduction**



Engineers use both physical and numerical models to predict the performance of communication antennas. Some of the numerical analysis programs in use are based on the Method of Moments and include NEC[1], MININEC[2], and Junction[3]. These programs are intended for modeling arbitrary geometries defined by wire frames and surface patches. They typically compute currents, charges, impedances, electric and magnetic fields, and other antenna parameters as output. Other numerical analysis programs are based on the Geometric Theory of Diffraction (GTD) and Finite Difference Time Domain (FDTD). GTD programs compute electric and magnetic fields from arbitrary geometries composed of generic shapes such as plates and cylinders. FDTD programs compute fields from partitioned volumes and surfaces.

Besides an analysis program, computer antenna modeling requires programs for inputting geometric, electromagnetic, and program control data and for analyzing and displaying results. Many of these support programs are sophisticated special-purpose tools. Others are off-the-shelf products like CAD programs, spreadsheets, and database management systems for inputting data, statistical analysis programs and graphics programs for analyzing output, and word processors and desk-top publishing systems for generating reports. These programs are often written in different languages and require input and output data to be in different formats.

Antenna models typically require large data sets for both input and output. Besides the antenna, all or part of the platform (ship, tank, jeep, airplane, etc.) may be required in the model. To model a ship may require thousands of nodes, wires, and patches. Many versions of the same model are often generated when investigating various antenna configurations. Output is often computed repeatedly over a range of frequencies for one or more potential antenna sites. Furthermore, Method of Moments codes have rigid requirements for model input. For example, there are restrictions on wire radius, wire segment length, wire spacing, number and angle of wires at junctions, size of surface patches, and the total number of unknowns (nodes, wires and patches). Therefore, users often have difficulty preparing and validating input data sets, especially when these sets are large. In addition, users may have difficulty converting data to the format required by the analysis software.

### **An Integrated Environment**

Many of these difficulties can be overcome by providing a single integrated environment for antenna modeling. A current Navy effort in this area is called the Numerical Electromagnetic Engineering Design System or NEEDS[4]. NEEDS will combine existing software tools into a single uniform environment. It will guide the user through the steps necessary to build a model, validate the model, compute currents and other EM parameters, and analyze and

display results. It will convert data as needed, store intermediate and final results, and document the history of models and their various versions. It will facilitate the reuse and sharing of model input and output. It will allow communication over networks for remote processing. In addition, the system will be portable across a variety of machine types and operating systems.

It is important to design NEEDS for maximum flexibility. It is inevitable that additional analysis and support tools will be added. It must be possible to modify the system easily in order to integrate new tools written in arbitrary languages. This requires that the major components of the system (model input, electromagnetic analysis, output display, process control, and data management) be implemented in separate independent modules. It must be possible to use the data in ways not necessarily anticipated at development time. Thus, data retrieval methods must be flexible. To support a variety of analysis codes, the model geometries should be represented by generic shapes from which wire frame or surface patch models can be automatically generated as required by the particular analysis codes used.

There are many issues to be addressed in building an integrated system such as NEEDS. These issues include conceptual model representation, partitioning algorithms for generating wire segments and patches, model evaluation and validation, and user interface designs. This paper, however, focuses on the issue of data management. This is an issue that becomes more important as databases increase in size and complexity. A significant amount of time and money is now being spent maintaining large databases that are basically incompatible. Methods are needed for handling large databases in ways that will allow for the efficient sharing of data among engineers and across tools and applications.

### **Data Management**

A major problem with integrating heterogeneous programs is the management of incompatible data sets. Each program in the system typically uses its own internal files and has its own unique file formats. Thus, there tends to be significant duplication of both data and data-accessing routines. This duplication is inefficient and can lead to serious inconsistencies in the data. In addition, conversion programs are needed to translate files from one format to another. One or more conversion programs may be needed between each pair of programs in the system. The number of conversion programs needed, therefore, generally increases as the **square** of the number of programs in the system. Finally, data files tend to be difficult to modify. Any change in the format of a file requires changes to all programs accessing that file.

One way to deal with these problems is to use a common database. By using a common database, each item of data needs to be

stored in only one place, and common data access routines can be provided. Furthermore, conversion programs are needed only between each program file and the database. Thus, the number of conversion programs needed will grow **linearly** with, rather than as the **square** of, the number of programs. This savings becomes more important as the system grows. Finally, as we shall see, database languages exist that make application codes less dependent on data file formats.

### Relational Databases

The leading database technology today is the relational database[5]. Relational databases are known for their ability to minimize data redundancy, provide flexible data representation, and allow for efficient data access. A relational database organizes data as a collection of tables. A separate table is used to represent each type of object or "entity" in the database. Each table has a fixed set of columns representing the characteristics or "attributes" of each object type. And each row of the table represents a single object or "instance" of that type.

For example, each node in a wire frame can be described by its x-, y-, and z-coordinates (Table 1). An additional attribute, called Node\_Number, is used as a key to identify each node uniquely.

NODES

Node_ Number	X	Y	Z
1	0.0	0.0	0.0
2	0.0	0.0	1.5
3	2.8	0.0	1.5
4	0.0	2.8	1.5

A List of Nodes  
Table 1

Tables also can be used to represent relationships among two or more entities. Relationship tables combine the key attributes (such as names or ID numbers) from two or more entities. For example, wires can be characterized by the indices of the nodes at each end of the wire (Table 2). Additional attributes, such as Radius and Number\_of\_Segments, can be added to describe the relationship further.

# WIRES

Wire_ Number	Node1	Node2	Radius	Number_Of_ Segments
1	1	2	0.001	4
2	2	3	0.01	2
3	4	2	0.002	4

A List of Wires  
Table 2

Note that the columns labeled "Node1" and "Node2" in the Wires Table contain values of the key attribute (Node\_Number) appearing in the Nodes Table. These attributes are called "foreign keys" in the Wires Table. A foreign key points to a key in another table.

It is important to design database tables carefully in order to reduce the number of blank spaces and the amount of redundant information. Redundant information wastes space and can lead to inconsistencies. A well-known process for designing relational database tables is called "normalization". Normalization breaks large tables into smaller tables so that each table describes a single atomic entity or relationship. Relationships among tables are preserved in the foreign keys. Smaller tables can later be combined into larger tables, as needed, by using an operation called a "join".

Both the Nodes Table and the Wires Table above are normalized. Notice that when multiple wires end at the same node, there is no need to repeat the x-, y-, and z-coordinates of that node. The coordinates are stored only once in the Nodes Table and a reference made to them in the Wires Table through the foreign keys. This helps to maintain the consistency of the data if changes are made to the node's coordinates.

Besides the database itself, a query language is needed for organizing, storing, and retrieving values from a database. The Structured Query Language (SQL) is an ANSI standard for relational databases. SQL commands can be executed interactively or embedded within a general-purpose programming language like C or FORTRAN. SQL queries allow the user to retrieve any subset of the rows of any table from any subset of its columns, as well as to combine tables. SQL provides flexibility in allowing the user to retrieve precisely the subset of data required. SQL is a non-procedural language. That is, it describes **what** subset of data to retrieve without describing **how** to retrieve it. This minimizes the amount of programming required. It also ensures that the code that accesses the data is independent of how the data is actually stored. This "physical data independence" ensures that changes to the physical structure of the database can be made without affecting the code

that accesses it. Some examples of SQL queries are as follows:

EXAMPLE 1. The query

```
SELECT SUM (Number_of_Segments)
FROM Nodes, Wires
WHERE (Node1 = Node_Number) AND (Z > 0.0);
```

retrieves the total number of segments on those wires for which the first endpoint has a positive z-coordinate. Notice that this query requires information from both the Nodes Table (for the z-coordinate) and the Wires Table (for the number of segments).

EXAMPLE 2. The SQL query

```
SELECT W1.Wire_Number, W2.Wire_Number
FROM Wires W1, Wires W2
WHERE ((W1.Node1 = W2.Node1)
      OR (W1.Node1 = W2.Node2)
      OR (W1.Node2 = W2.Node1)
      OR (W1.Node2 = W2.Node2))
AND (W1.Wire_Number < W2.Wire_Number);
```

finds each pair of adjacent wires (temporarily called "W1" and "W2") and retrieves their wire numbers. It combines a copy of the Wires Table with itself. The result is shown in Table 3 below.

Wire_Number	Wire_Number
1	2
1	3
2	3

Query Result  
Table 3

A complete database management system will provide capabilities other than simple storage and retrieval functions. These capabilities usually include multi-level security, backup and recovery in case of software or hardware failures, concurrency control to allow more than one user to access the database at the same time, and indexing and clustering to speed up access to the most frequently used data. Commercial database management systems usually provide tools for creating user interfaces that facilitate access to the database. And they allow database files to be imported from and exported to other operating system and database files.

## **The ORACLE Relational Database Management System**

The authors built a prototype of an integrated antenna modeling environment based on a relational database management system (RDBMS). The prototype demonstrates the feasibility of using a RDBMS for providing the following data management capabilities:

- ▶ a central uniform data repository
- ▶ efficient access to the data, either directly by the user or transparently through application programs
- ▶ low-level data validation
- ▶ conversion of file formats between those used by the database management system and those used by the application programs
- ▶ documentation of the history of models and their various versions and
- ▶ archiving of models for long-term storage.

Low-level data validation involves checking constraints on the data imposed by the conceptual model. Additional constraints on the data imposed by specific analysis programs are assumed to be handled by the analysis software itself. Security, concurrency, and network access were not considered to be important at this time.

We decided to use a commercial RDBMS for several reasons. First, commercial systems are extremely reliable. Second, the user interface tools provided by most commercial systems minimize the time needed to develop and update application software. And third, it would take many years to develop a system that would provide the same functionality as that currently provided by commercial systems.

There are many commercial RDBMS's on the market. ORACLE<sup>1</sup> was chosen for this project to achieve some standardization with other Army and Navy projects. ORACLE is a complete database management system providing all the capabilities mentioned above. It runs on a variety of machines under all major operating systems allowing application software to be easily ported. Versions of ORACLE are available that run over a network. ORACLE supports embedded SQL commands in both C and FORTRAN.

### **Description of the Prototype**

The prototype runs under MS-DOS<sup>2</sup> 3.3 and ORACLE RDBMS 5.1B on

---

<sup>1</sup>ORACLE is a registered trademark of Oracle Corporation.

<sup>2</sup>MS-DOS is a registered trademark of Microsoft, Inc.

an IBM-PC<sup>3</sup> or compatible with at least one hard disk drive and at least 256K of extended memory. The extra memory is required for the ORACLE RDBMS. The PC version of ORACLE is a single-user system.

The central data repository is a relational database with pre-defined tables. Users can directly access the database through SQL commands or indirectly through application programs and database forms. The database contains model input and output, model histories, control data, and some intermediate model-derived data.

In addition to the database, the prototype includes the following programs:

- ▶ the Junction code from the University of Houston for computing currents, charges, far fields, and near fields from models described by a combination of wires segments and triangular surface patches
- ▶ an ORACLE menu for navigating through the system
- ▶ ORACLE forms for inputting data and displaying results
- ▶ several C programs with embedded database queries for converting between the ORACLE database and Junction's ASCII formatted files and
- ▶ a supervisory program written in C.

Included in the Junction code are routines for producing surface patches from generic shapes such as cones, cylinders, and spheres.

During a typical modeling session, a user would

- ▶ define a new model or select an existing model for update
- ▶ input or update the model geometry and electromagnetic data including definitions of wires, surfaces, sources, frequencies, and far and near field locations, as desired.
- ▶ select the desired output such as currents, charges, far fields, and/or near fields
- ▶ request the system to compute the desired output
- ▶ view the output

The supervisory program ensures that programs are executed in the correct sequence and that each program receives its data in the proper format. The computation of output can be divided into several phases: 1) creating wire segments and surface patches, 2) computing additional geometry parameters such as the midpoint of wire segments and locations of body-wire junctions, 3) computing currents, and 4) computing charges, far fields and near fields, if desired. Each of these phases can be executed separately and intermediate results viewed. These phases must be executed in this

---

<sup>3</sup>IBM is a registered trademark of International Business Machines Corporation.

order, but not all changes to the input require recomputation of each phase. For example, a change in a source location requires that the currents, and hence charges and fields, be recomputed, but a change in a near field location requires only that near fields be recomputed. To control the execution sequence, tables in the database record which types of data have been defined by the user and which types of data have been computed by the system. By referring to these tables, the supervisory program prevents users from attempting to compute output data before all prerequisite input data have been entered, and it avoids computing data that are already available in the database.

The user traverses the system by means of a menu. Menu and sub-menu commands ultimately execute database commands, operating system commands, database forms, or other application software. A background menu, accessible from anywhere in the menu system, allows advanced users to execute their own operating system or database commands directly in order to accomplish specialized tasks.

A form-based user interface facilitates input and output of non-geometric data and geometric data for small models. Forms allow for the entry of data independent of the analysis software. Forms can do low-level data validation, such as type-checking and range-checking, before data is committed to the database. This helps to maintain the integrity of the database, and it provides immediate feedback to the user when mistakes are made. Help messages and default values can be provided for each field in the form. The developer can designate certain fields as key fields which must have unique values. Non-key fields can be designated as either mandatory or optional. (Optional fields may contain null values.) And user updates can be restricted to a subset of the available fields, if desired.

Since these forms are closely integrated with the database, they can perform functions not common to other forms software. For example, forms can be designed to make simultaneous updates to other fields, such as foreign keys, in other database tables (see Example 2 below). This helps to maintain the consistency of the database. The same form that is used to insert or update data in the database can be used to retrieve data from the database. The user performs queries by providing a value or a range of values in one or more fields of the form. The form will then retrieve all database records that include those field values. To allow additional functionality, database operations and/or C code can be tagged to certain events (such as updates, queries, or cursor moves) that are executed when the user causes those events to occur. These "triggers" are used for more advanced data validation and housekeeping functions.

The following are some examples of the capabilities of database forms. The form in Figure 1 references the Nodes and the



Wires Tables discussed above.

NODES			
Node Number	X (meters)	Y (meters)	Z (meters)
1	0.0	0.0	0.0
2	0.0	0.0	1.5
3	2.8	0.0	1.5
4	0.0	2.8	1.5

WIRES				
Wire Number	Node 1	Node 2	Radius (meters)	Number of Segments
1	1	2	0.001	4
2	2	3	0.01	2
3	4	2	0.002	4

A Form for Inputting and Displaying Wires  
Figure 1

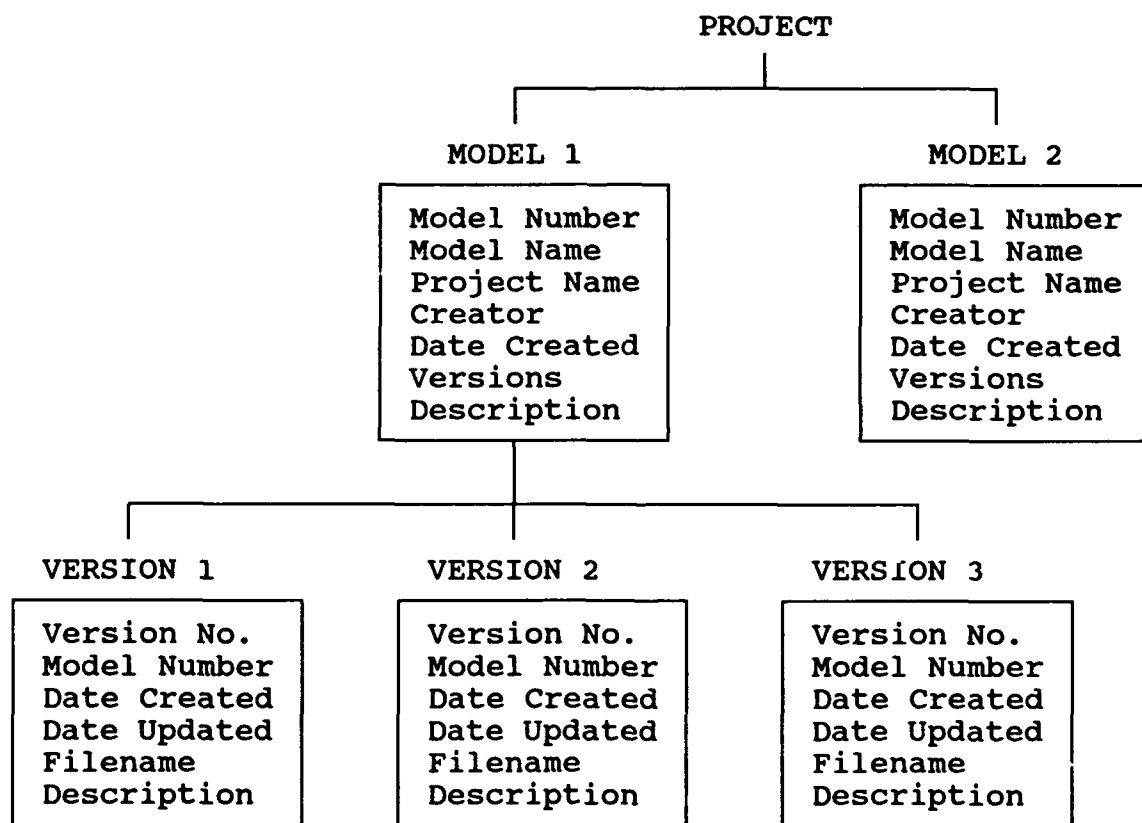
**EXAMPLE 1.** To insert a new wire, the user types a value for each field in the Wires Table and invokes the Insert function. The form can verify that the two nodes referenced by the new wire have already been defined in the Nodes Table.

**EXAMPLE 2.** If the user wishes to change the index of Node 4 to Node 5, the user changes the "4" to a "5" under Node Number in the Nodes Table and invokes the Update function. The form can propagate this change automatically to the Wires Tables, so that Wire 3 would then connect Node 5 to Node 2.

**EXAMPLE 3.** To list all the Nodes in the database with an x-coordinate greater than 0.5 meters, the user types ">0.5" in the X field of the Nodes Table and invokes the Query function.

The prototype also records the history of models as they are developed by the user. It allows the user to document different

projects, several models belonging to each project, and multiple versions of each model (see Figure 2). Each model is described by a model number, model name, project name, the name of the person creating the model, the date it was created, the number of versions it has, and a textual description of the model. Each version is described by a version number, the number of the model to which it belongs, the date it was created, the date it was last updated, the name of the file where it is stored, and a textual description. A new version is created automatically after the user computes the currents for the existing version. Thus, each time the currents are computed for a group of frequencies, the input data become fixed for that version. Any additional changes to the input data are reflected in the new version.



The Description of Model Histories  
Figure 2

Because of the limited memory of a PC, it would be difficult to store input and output data for all versions in the database simultaneously. Therefore, the database contains workspace for the current version only, plus a catalog of past and current models and versions. In addition, alternative sources, near field regions, and far field regions can be stored in the database. Previous models and their various versions are archived on disk and can be reloaded from the on-line catalog as needed.

## Lessons Learned

A relational database management system proved to be well-suited to this application. It was possible to develop a fairly complex, modular system in a short time due to the database menu- and form-building software and the use of embedded SQL commands in application programs. These application-building tools also allow the developer to provide the user with transparent access to database. Thus, the typical user does not need to know that a relational database exists or how it is organized.

ORACLE, however, is primarily a business-oriented database management system. Therefore, its query languages lack a complex data type and such built-in functions as square roots, exponentials, logarithms, and trigonometric functions needed for scientific and engineering applications. Thus, embedded C code is needed to perform these functions. A scientific database management system would provide these as extensions of the SQL language.

Another disadvantage of the ORACLE RDBMS for some is its size and cost. The database management system requires at least 1 MB of main memory and about 9 MB of disk space in addition to disk space for the database itself. Large amounts of memory and disk space, however, are becoming much more affordable. And ORACLE's portability and network capabilities may ultimately far outweigh these disadvantages.

The most significant advantage of the relational database is the flexibility that it provides. It allows data to be combined and retrieved in many ways. As the modeling capabilities of the system are extended in the future (to include, for example, loads, transmission lines, or material types), it will be necessary to add new tables or new columns of existing tables to the database. This database extension, however, does not require changing either existing data or existing code. As long as table and column names are not changed, existing analysis programs and support tools will still execute properly on the modified database.

It was difficult to keep more than one version of a large model in the database at one time due to the memory limitations of a PC. Therefore, the ability to make arbitrary comparisons of data across different versions or models is not a built-in capability of the system at this time. For advanced users, however, it is possible to make such comparisons by accessing the database directly. This requires knowledge of simple SQL commands in order to create new tables and to move data from one table to another.

Any environment that generates so much data, however, should provide for an efficient way to browse through the data, compare data from different models and versions, and do other kinds of data synthesis. In the future it would be advisable to run the database

management system on a dedicated workstation or minicomputer as a database server, so that numerous models and their various versions could be kept in the database at one time. A more sophisticated archival system is needed for expanding the work area as needed and for automatically storing data on disk or tape as available memory is used up.

### **Future Possibilities**

It is anticipated that additional tools will be added to the integrated system in the future. These include graphical input programs for large geometries, 2- and 3-dimensional graphical output display programs, a windows-based user interface, and additional EM analysis tools, such as other Method of Moments (MOM) codes, Geometric Theory of Diffraction (GTD) codes, and Finite Difference Time Domain (FDTD) codes. Since this prototype was developed, ORACLE Corporation has marketed an interface for Microsoft Windows<sup>4</sup> 3.0 that will also allow developers and users to access ORACLE data from most Windows applications.

Different analysis codes (MOM, GTD, and FDTD) require slightly different input in different formats to model the same physical objects. In the past, as new antenna analysis programs were developed, new special-purpose tools were written to support model input. Thus, there is a need for tools that input and store generic objects that can be adapted to any analysis tool. This would allow input objects to be used with many different codes in many different applications. These codes could be modified to access the database directly thereby eliminating the need for file format conversion programs. Substantially the same tools are also needed to display such output as currents, charges, far fields, and near fields regardless of the program that generates this output. This need for reusable tools will become more important as the tools become more sophisticated and as integrated systems become larger. A common database format for describing geometries would facilitate the realization of this goal.

Finally, advantages can be foreseen for using standard databases and standard query languages in order to interface with other modeling systems. For example, data generated numerically on the computer could be more easily compared with data measured from physical antenna models if a common data format were used. This also would facilitate the integration of software for antenna modeling with software such as COEDS[4] for modeling entire communication systems. The use of a relational database management system that is widely portable can be an important step toward creating sharable data among such related applications.

---

<sup>4</sup>Microsoft Windows is a trademark of Microsoft Corporation.

### References

1. Burke, G. J. and A. J. Poggio, "NEC - Method of Moments ", Naval Ocean Systems Center Technical Document 116, 1981, revised 1988.
2. Rockway, J. W., J. C. Logan, D. W. S. Tam, and S. T. Li, ***The MININEC System: Microcomputer Analysis of Wire Antennas***, Artech House, Inc., 1988.
3. Wilton, D. R. and S. U. Hwu, "Junction Code User's Manual", Naval Ocean Systems Center Technical Document 1324, August 1988.
4. Li, S. T., J. C. Logan, and J. W. Rockway, "Ship EM Design Technology," ***Naval Engineers Journal***, May 1988, pp. 154 - 165.
5. Korth, H. and A. Silberschatz, ***Database System Concepts***, McGraw-Hill, 1986.

# **An Application of the Hybrid Moment Method/Green's Function Technique to the Optimization of Resistive Strips**

R. Craig Baucke  
GE Aircraft Engines M/D J185  
1 Neumann Way  
Cincinnati, OH 45215-6301

**Abstract** - *An automatic method of synthesizing resistive tapers is developed. This method embeds a hybrid moment method/Green's function inside a nonlinear optimization package. Using this technique, resistive tapers are rapidly synthesized for complex scatterers which can consist of multiple resistive strips, as well as large, arbitrary conducting regions. The method is applied to the optimization of resistive tapers that reduce the diffraction from conducting scatterers.*

## **I. INTRODUCTION**

Resistive strips are used in various applications to modify the electromagnetic scattering characteristics of an antenna or scatterer. They are used to reduce the diffraction from conductive edges or discontinuities [1-2], approximate an infinite ground plane [3], improve the performance of a compact range reflector [4], and attenuate energy in waveguide [5]. Even greater control over the scattering characteristics of the structure are obtained by tapering the value of the strip resistance [6,7].

While a physical-optics based method for synthesizing effective resistive tapers has been developed by Haupt [8], it is only applicable for a simple single strip geometry at E-polarization. The more general problem of defining optimum resistive tapers for multiple strips in the presence of arbitrary scatterers for arbitrary polarization is usually done in a trial and error fashion using established tapers as a starting point. The effectiveness of the proposed taper is computed using the method of moments (MoM or MM) or other numerical method (or measurements). While the trial and error approach to resistive taper design has led to a database of "good tapers", this approach is slow and may not result in the optimum taper, if one exists.

In this work, an automated method of synthesizing optimal resistive tapers is developed. Resistive tapers are computed in a minimum amount of

time within certain physical constraints. In effect, the trial and error method is replaced by a nonlinear optimization technique which searches for an optimal solution. The result is improved taper performance and drastically reduced design times. To implement this concept, the hybrid MM/Green's function (HMGF) technique described in [9] is applied to a moment method which analyzes two dimensional conductive and resistive strips at both E-polarization ( $TM_z$ ) and H-polarization ( $TE_z$ ). This approach is then encapsulated within a nonlinear optimization program such as [10, 11]. The resulting method rapidly computes the scattering levels from a number of different resistive tapers and searches for an optimum configuration within user defined constraints. This paper shows that the optimization of moment method analysis is only practical due to the application of the HMGF technique. In addition, the simultaneous optimization of a resistive taper for both polarizations is demonstrated, as well as the optimization of multiple tapers simultaneously.

## II. MOMENT METHOD APPROACH

The choice of moment methods is critical to the efficiency of the optimization process. For E-polarization, an efficient Galerkin method developed by the author which utilizes pulse basis and pulse testing functions for metal and resistive scatterers is chosen. For H-polarization, the method of Liu and Balanis [12] has been enhanced to include resistive strips. This method uses pulse basis functions and thereby creates fictitious line charges at cell boundaries. In practice, the method of [12] provides fairly accurate far field results as long as the cell widths are about 1/10th of a wavelength. Since point matching is employed, it is very efficient.

Both the E-polarization and H-polarization moment methods use the electric field integral equation (EFIE). While the EFIE allows for the analysis of open structures such as resistive strips, interior resonances may exist for closed perfectly conducting (PEC) structures.

The EFIE for each moment method is developed by relating the incident electric field  $E^i$  to the total field  $E^t$  and the scattered field  $E^s$  by

$$(1) \quad E^i(x,y) = E^t(x,y) - E^s(x,y).$$

For resistive strips, the total field is defined as

$$(2) \quad E^t(x,y) = R(x,y) J(x,y)$$

where  $R$  is the surface resistance and  $J$  is the surface current density.

Substituting (2) into (1) and applying basis and testing functions yields

$$(3) \quad \langle E^i, B_m \rangle = \langle RJ_n P_n, B_m \rangle - \langle E^s, B_m \rangle$$

where  $P_n$  is the basis function for the  $n$ th source cell,  $B_m$  is the testing function for the  $m$ th test cell and  $J_n$  is the unknown current on the  $n$ th cell. The first inner product on the right side of (3) contains the resistive term, and is defined as

$$(4) \quad \Delta Z = \langle RJ_n P_n, B_m \rangle.$$

The term  $\Delta Z$  is nonzero when the domains of the basis and testing functions overlap, and when the resistance of the source domain is nonzero.

Pulse basis and testing functions are applied for  $TM_z$  polarization. In this case,

$$(5) \quad \Delta Z = \langle RJ_n \Pi_n, \Pi_m \rangle$$

where  $\Pi$  is the pulse function. This expression is nonzero when  $m=n$  and contributes only to the diagonal terms of the moment matrix.

Pulse basis and point matching are applied for  $TE_z$  polarization. In this case,

$$(6) \quad \Delta Z = \langle RJ_n \Pi_n, \delta_m \rangle,$$

which is also nonzero only for diagonal elements of the matrix. Since (5) and (6) are nonzero only for diagonal elements of the moment matrices, when the resistances of the cells are modified, only the diagonal terms of the impedance matrix are changed, and only by a constant value. If other basis functions such as piecewise linear (triangular) or sinusoidal are chosen,  $\Delta Z$  is nonzero for off-diagonal terms, and more matrix modification is required when resistance values are changed.

Once the basis and testing functions are applied, (3) reduces to a matrix equation  $Zx=B$ . At this point, the HMGF method is applied to the system matrix.



### III. APPLICATION OF HYBRID MM/GREEN'S FUNCTION TECHNIQUE

To apply the HMGF method, the scatterer is divided into two sections. One of the sections contains the portion of the scatterer where the resistance could be modified in the optimization process. This section will be referred to as scatterer 1 (or S1). The rest of the scatterer (the portion remaining unmodified) is called scatterer 2 (or S2). This is shown in figure 1.

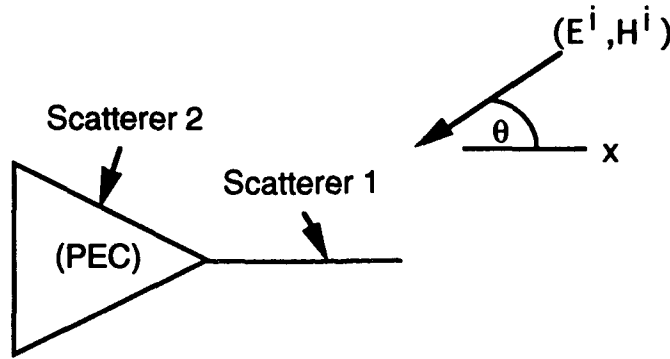


Figure 1. An example: scatterer 1 and scatterer 2.

In figure 1, S1 is a resistive sheet and S2 is a perfectly conducting closed triangular cylinder. The resistance of each cell on S1 may be modified while S2 will remain unchanged. The system matrix  $Z$  is partitioned as

$$(7) \quad Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

where  $Z_{11}$  contains the matrix elements in which the observation and source points are located on S1,  $Z_{12}$  contains the elements in which the observation point is on S1 and the source point is on S2,  $Z_{21}$  contains the elements with the observation point on S2 and the source point on S1, and  $Z_{22}$  contains the elements with the observation and source points on S2. The order of  $Z_{11}$  is  $N_1$ , where  $N_1$  is the number of cells on S1, and the order of  $Z_{22}$  is  $N_2$ , where  $N_2$  is the number of cells on S2.

Performing the linear algebra described in [9], the matrix equation is reduced from order  $N_1 + N_2$  to order  $N_1$  and reformulated as

$$(8) \quad (Z_{11} - Z_{12}Z_{22}^{-1}Z_{21})I_1 = B_1 - Z_{12}Z_{22}^{-1}B_2$$

where  $B_1$  is the excitation vector for S1,  $B_2$  is the excitation vector for S2, and  $I_1$  is the current solution for S1. The solution current on S2 can be found from

$$(9) \quad I_2 = Z_{22}^{-1}B_2 - Z_{22}^{-1}Z_{21}I_1$$

The monostatic and bistatic scattering from the combination of S1 and S2 can be computed from the solution currents  $I_1$  and  $I_2$ .

#### IV. APPLYING NONLINEAR OPTIMIZATION

In order to apply nonlinear optimization to this method, the echo width of the scatterer is computed by solving (8) with the initial (first guess) resistance values. This computation is three to four times slower than analyzing the entire scatterer with a traditional moment method approach, due to the computation of the complete  $Z_{22}$  inverse and several matrix multiplies. At this point, the inverse of the  $Z_{22}$  matrix is stored, as well as the  $Z_{12}$  and  $Z_{21}$  matrices. These matrices will not change in the following optimization iterations.

Once the initial resistive configuration is analyzed, the nonlinear optimizer computes new resistive values for the cells on S1. The system matrix in (8) is recomputed by subtracting and adding values to the appropriate diagonal terms. When the system in (8) is solved during subsequent iterations, the matrix fill time is very small and the solution time is drastically reduced compared to traditional moment methods. A comparison of the CPU time required for optimization of this method and a traditional moment method are shown in figure 2. These performance figures demonstrate the feasibility of performing nonlinear optimization on moment method calculations.

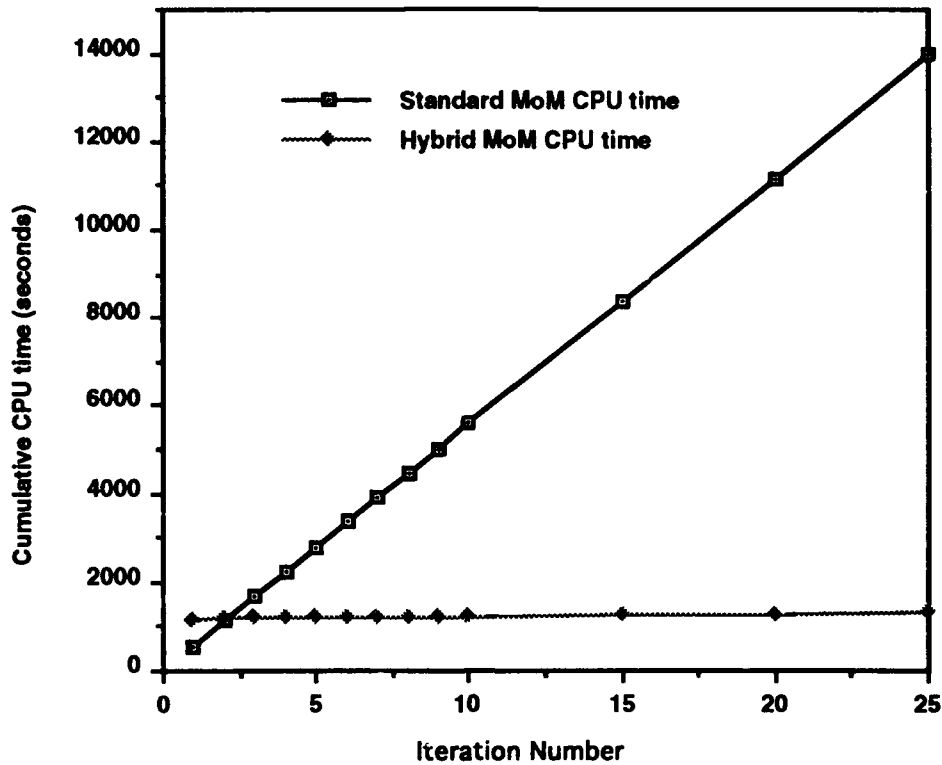


Figure 2. Comparison of the CPU time required for optimization with and without applying the HMGF technique.

The operation of the nonlinear optimizer requires the definition of a penalty function, which is the function to be minimized. While this function could be related to any quantity computed by the moment method (surface currents, near fields, etc.), in this case the penalty function  $F$  is defined as

$$(10) \quad F = \sum_{\alpha=1}^2 \frac{1}{N_{\theta}} \sum_{i=1}^{N_{\theta}} \sigma^{\alpha}(\theta_i)$$

where  $\alpha=1$  is  $TM_z$  polarization,  $\alpha=2$  is  $TE_z$  polarization,  $N_{\theta}$  is the number of monostatic angles,  $\sigma$  is the echo width in wavelengths, and  $\theta_i$  is the  $i$ th monostatic angle. The function  $F$  is essentially a sector average of the echo width.

The optimizer attempts to minimize  $F$  by modifying the resistive values on  $S_1$ . A flow chart of the optimization process is shown in figure 3.

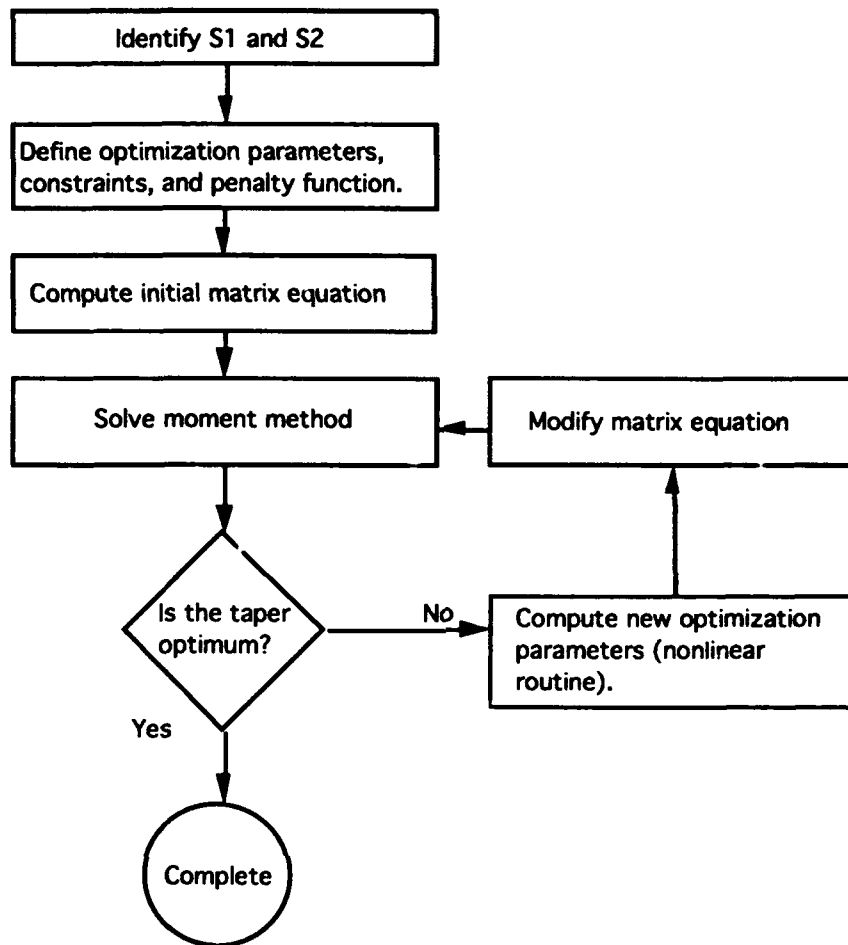


Figure 3. Flow chart of the resistive taper optimization process.

The choice of nonlinear optimization routines is limited by the type of function to be minimized. Since gradients of the function to be minimized in this work (10) are difficult to obtain, optimization methods which do not require explicit gradients are chosen. In addition, the resistance values of the cells are constrained to positive values that can be manufactured (it is difficult to accurately produce and measure resistive strips with resistances much greater than 3000 ohms/square). Thus, a constrained optimization is chosen.

In this work, the rotating coordinates method described in [10] and the Complex algorithm [11] are both applied. Both are direct search routines which utilize different search strategies. As in the case of all nonlinear optimizers, convergence to the absolute minimum is not guaranteed. To avoid nonoptimal local minima, the optimization process is carried out with different starting points and step sizes.

## V. RESISTIVE TAPER OPTIMIZATION APPROACHES

The method developed in this paper is applied to the optimization of resistive tapers on S1 in two different fashions. In the first, the resistance value of each cell is an optimization parameter, while in the second approach the coefficients of a polynomial function are the parameters.

The first approach provides a very flexible approach to taper optimization, but often results in resistive tapers which cannot be constructed due to large fluctuations in the values of the resistances along the strips. These fluctuations in the optimal taper can be reduced by constraining the resistive values properly. However, it may still be difficult to create a smooth taper by optimizing in this fashion, and the effective bandwidth of the taper (the frequency range over which the taper has acceptable performance) may be small.

In the second approach, polynomial optimization, the nonlinear optimizer adjusts the coefficients of polynomials over each resistive taper in S1. For instance, if the user chooses a quadratic polynomial taper,

$$(11) \quad R(x) = ax^2 + bx + c \quad (0 \leq x \leq 1),$$

coefficients  $a$ ,  $b$  and  $c$  are modified by the optimizer. The variable  $x$  is the normalized distance along the strip.

The natural result of this optimization is a smooth taper (with the smoothness dependent on the order of the polynomial), as well as improved bandwidth. However, the performance at the frequency of optimization will probably be inferior to the first method. Thus, the first method is better for narrow band designs, and the second method is better for broadband designs. A comparison of the echo width and optimal taper computed by these two approaches for an example problem consisting of a flat resistive strip in front of a PEC is given in figures 4 and 5.

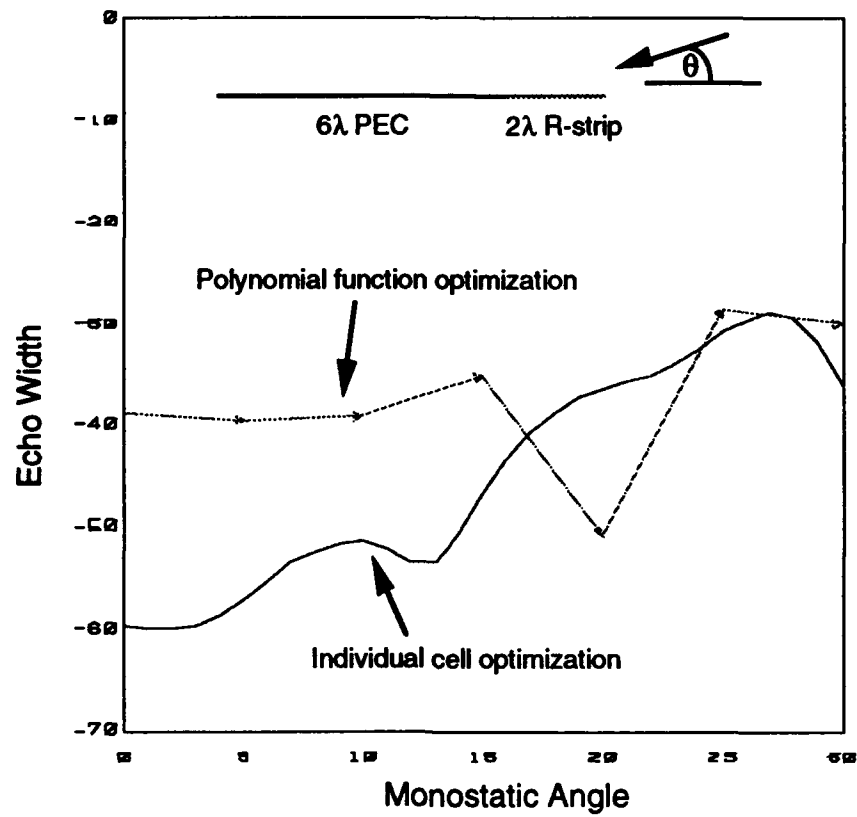


Figure 4. Comparison of echo width for resistive strip optimization using individual cell optimization and polynomial function optimization.

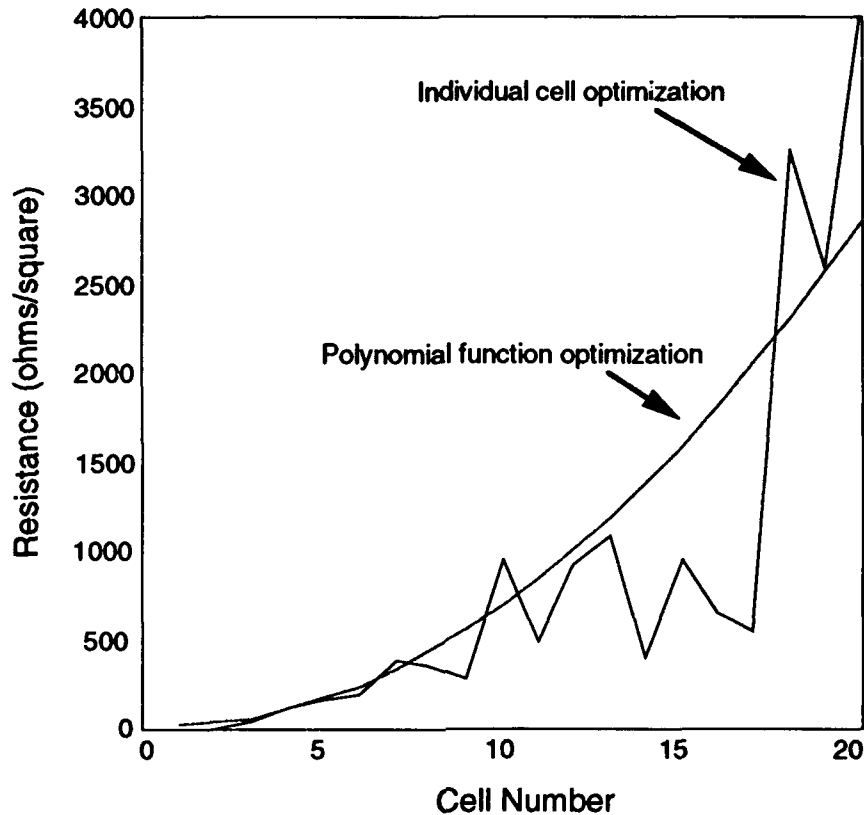


Figure 5. Comparison of resistive taper functions for resistive strip optimization using individual cell and polynomial function optimizations.

In figure 5, the optimal polynomial and individual cell values are shown as smooth and piecewise linear functions, respectively. In the context of the moment methods used in this work (5-6), the resistance of each cell is defined as a constant. Therefore, the functions in figure 5 are approximated by a set of step functions in the moment method discretization.

## VI. OPTIMIZATION RESULTS

The first of two simple example problems is shown below.

$2\lambda$ Resistive strip	$3\lambda$ PEC strip	$2\lambda$ Resistive strip
----------------------------	----------------------	----------------------------

---

Figure 6. A single resistive strip scatterer.

In this problem, the leading and trailing resistive strips are optimized at both  $TM_z$  and  $TE_z$  polarizations using a quadratic taper that is constrained to resistive values between 3000 and 1 ohms/square. The initial values of each resistive strip are 200 ohms/square with no variation along the strips. The resistive strip scatterer S1 consists of 40 cells (20 cells on the leading edge and 20 cells on the trailing edge), and the metal scatterer S2 is divided into 30 cells. Figure 7 compares the echo widths of the optimized scatterer to the initial scatterer for  $TM_z$  and  $TE_z$  polarization. The penalty function in both problems is the average monostatic echo width of the scatterer from +30 to -30 degrees, sampled every 5 degrees.



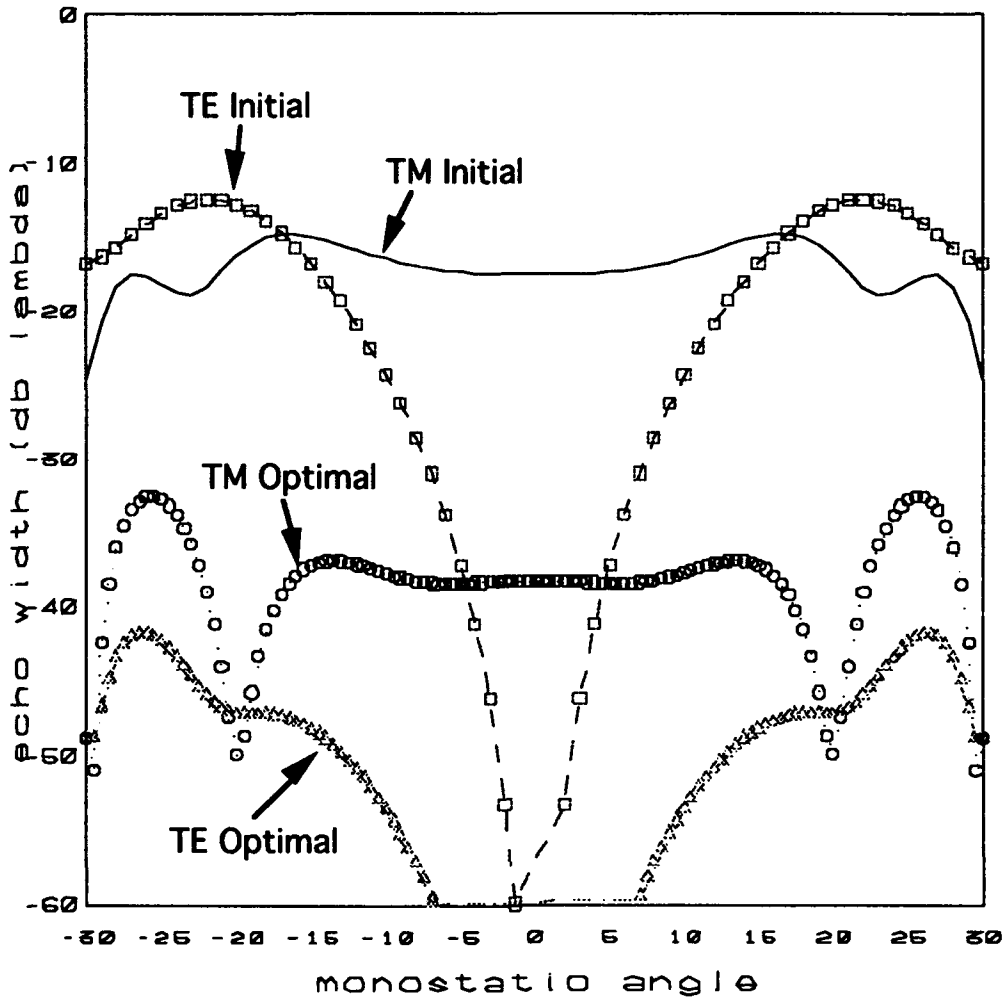


Figure 7. A comparison of the initial and optimized monostatic echo width from the scatterer in figure 6.

The resulting optimal quadratic taper on the leading edge strip is

$$(12) \quad R(x) = 3000x^2 + 131x + 6$$

and the optimal taper on the trailing edge strip is

$$(13) \quad R(x) = 2778x^2 + 4x + 10.$$

where the zero value of  $x$  in (12) and (13) is located at the junction of the corresponding resistive strip and PEC.

The optimization results in figure 7 require 496 iterations of the resistive values and 150 CPU seconds on a VAX 6510.

Both of the optimization techniques [10, 11] have been applied to this example, yielding nearly identical results. The results in figure 7 are obtained using the optimization method of [10].

In the second problem, another identical strip scatterer is placed  $0.75$  wavelengths above the original scatterer as shown in figure 8.

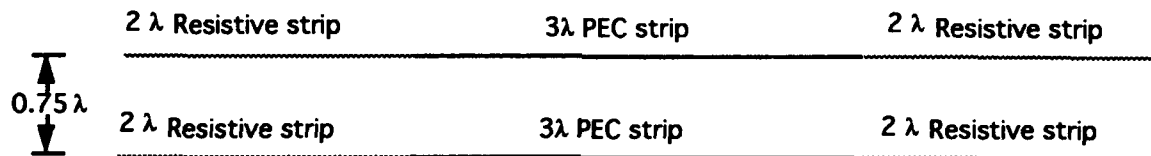


Figure 8. Two parallel resistive strips.

In the scatterer above, the four resistive strips making up S1 are divided into 80 cells and the PEC strips making up S2 consist of 60 cells. The same penalty function is used as in the previous example.

Figure 9 compares the echo widths of the optimized scatterer in figure 8 to the initial scatterer for  $TM_z$  and  $TE_z$  polarization.

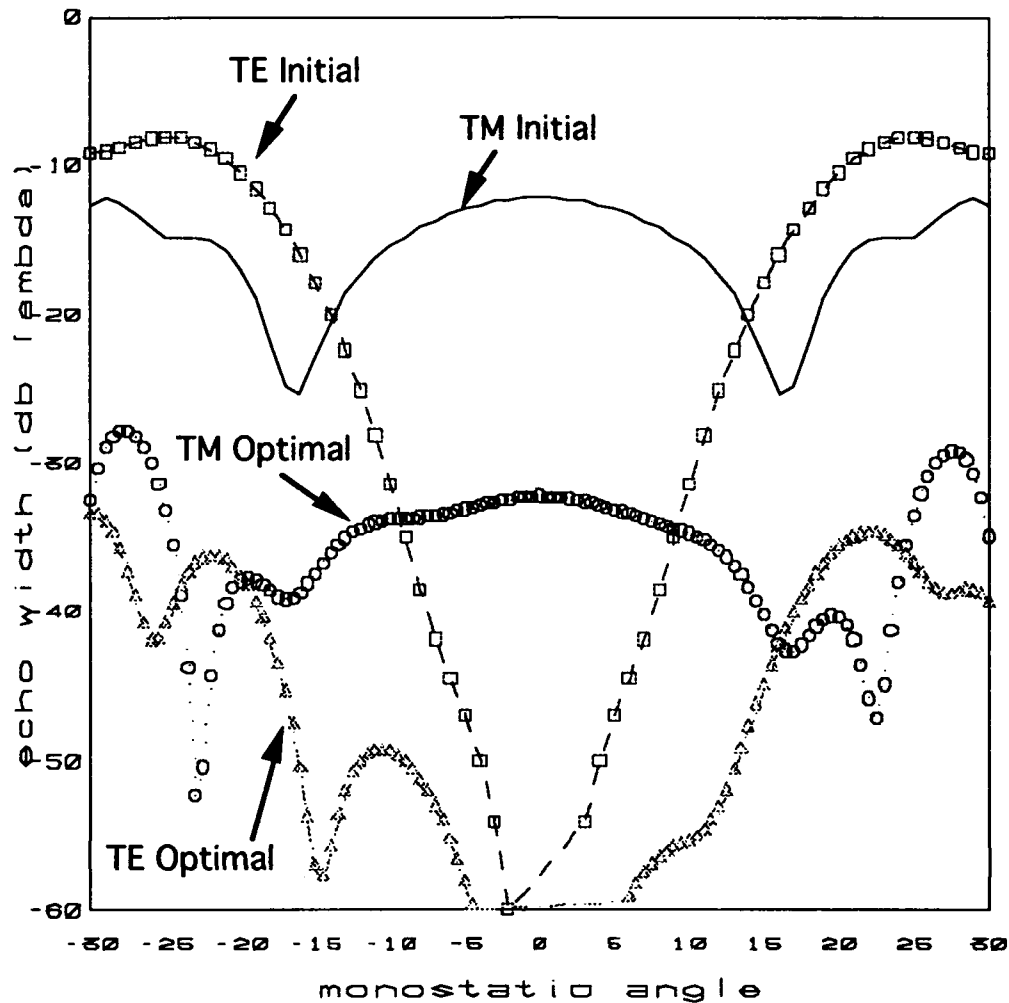


Figure 9. A comparison of the initial and optimized monostatic echo width from the scatterer in figure 8.

The resulting optimal quadratic taper on the leading edge strips are

$$(14) \quad \begin{aligned} R(x) &= 3000x^2 + 4x + 1 \text{ (upper)} \\ R(x) &= 2990x^2 + 4x + 3 \text{ (lower)} \end{aligned}$$

and the optimal taper on the trailing edge strips are

$$(15) \quad \begin{aligned} R(x) &= 1975x^2 + 12x + 12 \text{ (upper)} \\ R(x) &= 1065x^2 + 14x + 4 \text{ (lower).} \end{aligned}$$

The optimization results in figure 9 require 430 iterations of the resistive values and 534 CPU seconds on a VAX 6510.

## VII. CONCLUSIONS

In this work, an automatic method of synthesizing optimum resistive tapers for multiple arbitrary resistive strips in any scattering environment has been developed. The optimization is performed simultaneously for both  $TE_z$  and  $TM_z$  polarizations. The results of individual cell and polynomial function optimization are demonstrated and compared. The efficiency of this method due to the application of the HMGF technique is shown. While two dimensional results are shown here, this method could be easily applied to a three dimensional moment method structure.

## REFERENCES

- [1] T.B.A. Senior, "Scattering by resistive strips," *Radio Science*, vol. 14, no. 5, Sept.-Oct. 1979, pp. 911-924.
- [2] M.C. Heaton, "Electromagnetic scattering from impedance strips and impedance-loaded conducting strips," Master's Thesis, Air Force Institute of Technology, Dec. 1990.
- [3] G.F. Herrmann, "Numerical Computation of Diffraction Coefficients," *IEEE Trans. Antenna Propagat.*, vol. 35, no. 1, January 1987, pp. 53-61.
- [4] P.A. Beyerle, I.J. Gupta and E.H. Newman, "Compact range reflectors with resistive strip terminations - TM case," Ohio State University Technical Report 721223-5, Sept. 1989.
- [5] I.M. Braver, P. Sh. Fridberg and K.L. Garb, "Frequency-independent waveguide attenuator with a resistive film in the diagonal plane," *Microwave Journal*, Nov. 1990, pp. 179-182.
- [6] T.B.A. Senior and V.V. Liepa, "Backscattering from Tapered Resistive Strips," *IEEE Trans. Antenna Propagat.*, vol. 32, no. 7, July 1984, pp. 747-751.
- [7] R.L. Haupt and V.V. Liepa, "Resistive tapers that place nulls in the scattering patterns of strips," *IEEE Trans. Antennas Propagat.*, vol. 38, no. 7, July 1990, pp. 1117-1119.
- [8] R.L. Haupt and V.V. Liepa, "Synthesis of tapered resistive strips," *IEEE Trans. Antenna Propagat.*, vol. 35, no. 11, Nov. 1987, pp. 1217-1225.

- [9] E.H. Newman, "An overview of the hybrid MM/Green's function method in electromagnetics, " *Proceedings of the IEEE.*, March 1988, pp. 270-282.
- [10] H.H Rosenbrock, "An Automatic Method for finding the Greatest or Least Value of a Function, " *Computer Journal*, vol. 3, October 1960, pp. 175-184.
- [11] P.E. Gill, W. Murray and M.H. Wright, *Practical Optimization*, Academic Press, New York, 1981.
- [12] K. Liu and C.A. Balanis, "Simplified formulations for two-dimensional TE-polarization field components, " *IEEE Trans. Antenna Propagat..*, vol. 39, no. 2, February 1991, pp. 259-262.

# Parallel Matrix Solvers for Moment Method Codes for MIMD Computers

David B. Davidson  
Dept. of Electrical and Electronic Engineering  
University of Stellenbosch  
Stellenbosch 7600  
South Africa.  
E-mail:davidson@firga.sun.ac.za

## Abstract

Parallel algorithms are presented that are suitable for the solution of the system of linear equations generated by moment method problems on local memory Multiple Instruction, Multiple Data (MIMD) parallel computers. The two most widely used matrix solution algorithms in moment method codes are described, namely the conjugate gradient (CG) method and LU decomposition. The underlying philosophy of parallelism is briefly reviewed. Suitable parallel algorithms are then described, presented in pseudo-code, their timing behaviour analyzed theoretically, and timing results measured on a particular MIMD computer — a transputer array — are presented and compared to the theoretical timing models. It is concluded that efficient parallel algorithms for both the CG and LU exist and that MIMD computers offer an attractive computational platform for the solution of moment method problems with large numbers of unknowns.

<i>Symbol</i>	<i>Definition</i>
$t_{comm}$	Time to send one complex word between adjacent processors.
$t_{calc}$	Time for a real floating point + or $\times$ .
$\beta$	The ratio $t_{comm}/t_{calc}$
$M$	Number of unknowns (dimension of the matrix).
$N$	Number of processors.
$d$	Depth of the binary tree.

**Table 1:** List of symbols used frequently in this paper.

## 1 Introduction

### 1.1 Background

It had long been accepted that the applicability of the moment method is limited by available computational capability, in particular memory and speed of computation [1]. For a problem with no special properties such as symmetry as a result of reflection, rotation, or translation, the

Notation	Definition
$[A]$	The matrix $A$ .
$[A]^T$	The Hermitian (complex conjugate) transpose of matrix $A$ .
$a_{i,j}$	The $ij$ -th element of matrix $A$ .
$[x]$	The (algebraic) vector $x$ .
$x_i$	The $i$ -th element of vector $[x]$ .
$  [x]  $	The Euclidean norm of the vector $[x]$ of length $n$ ; $  [x]   = \sum_{i=1}^n  x_i ^2$ .
$ x $	Absolute value of scalar $x$ .
$[x]$	The ceiling function of $x$ , i.e. the smallest integer $\geq x$ .
$\wedge$	The Boolean AND operation.
$\text{mod}(a)$	The modulo( $a$ ) operator.
$O(M^n)$	Of the order of $M^n$ .

**Table 2:** Notation used in this paper.

computer time requirement grows at least as the cube of the number of unknowns, which is at least linearly related to the electromagnetic size (length, surface area or volume, depending on the particular problem and formulation) of the structure being simulated. The matrix equations to be solved are in general complex, non-symmetric and full, although certain formulations — and also physical symmetries, if present — may yield matrices with more structure. For problems which are not small electromagnetically, this presents formidably large systems of linear equations that must be filled and solved. The emergence of *vector supercomputers* has permitted the solution of much larger problems than could previously be handled. These computers, epitomized by the CRAY series, the first of which was installed in 1976, represented a tremendous increase in computational resources for researchers with access to one. However, such systems are extremely expensive, and not readily available outside the U.S.A., Europe and Japan at the time of writing. There are also limits on the computational speed of such systems. This paper considers the use of a different type of computer, the local (also known as distributed) memory Multiple Instruction Multiple Data (MIMD) computer; the algorithms described in this paper were run on an array of INMOS T800 transputers, an example of such an array. Such MIMD systems offer performance potentially rivaling that of the vector supercomputers, but require that the algorithms be very carefully designed to exploit the parallel architecture and thus obtain something approaching the manufacturer's claimed peak performance.<sup>1</sup> This paper concentrates on the derivation, analysis, implementation and testing of such algorithms, for the conjugate gradient (CG) and LU matrix solvers, and is an extension of previous papers by the author [3, 4].

<sup>1</sup>The transputer array used in this paper does not deliver performance on par with conventional supercomputer systems such as the CRAY machines already mentioned. However, in the light of the next generation of massively parallel arrays — with hundreds or thousands of processors compared to the dozens used in this paper, and with each processor running far faster than the transputers used here — the conventional supercomputer “now seems poised for an indefinite but inexorable decline” [2, p.27].

## 1.2 Parallel Processing

The fundamental principle underlying parallel (or concurrent) processing is that once the limits on speed imposed by a certain computing technology have been reached, the most obvious way of building a faster computer is to perform operations simultaneously. Two fundamental ways of implementing parallelism have emerged, namely pipelining and replication. The former involves overlapping parts of operations in time and is the approach taken by the vector supercomputers; the latter provides more than one functional unit (e.g. CPU), permitting operations to be performed simultaneously, and is the approach taken by such systems as arrays of transputers or i860 processors. The historical background of parallel computers and a more detailed explanation of pipelining and replication may be found in the author's tutorial paper [3], and with minor revisions in [5, Chapter 3].

Several methods have been proposed to characterize parallel computers, but the most widely used are speed-up and efficiency. Speed-up,  $S$ , is the ratio of time taken by an equivalent serial algorithm running on one processor,  $T_s$ , to the time taken by the parallel algorithm using  $N$  processors,  $T_p$ . Efficiency,  $\epsilon$ , is the speed-up normalized by the number of processors. Formally,

$$S = \frac{T_s}{T_p} \quad (1)$$

$$\epsilon = \frac{S}{N} \quad (2)$$

$S$  is usually bounded from above by  $N$  and  $\epsilon$  is hence usually bounded from above by 1 — although under very special circumstances an efficiency exceeding 1 is at least theoretically possible [5, Section 3.4.1]. Speed-up is the fundamental issue of importance for the user — it states how much faster his algorithm will run on  $N$  processors than on one. Efficiency is self-evident. The most important requirement for a parallel program — other, obviously, than its correctness — is to obtain the maximum possible speed-up, and thus also efficiency, from the available parallel hardware.

At present a major effort is required by the user to properly exploit parallel processing, in particular for MIMD systems. Automatic vectorizing compilers have simplified the task for pipelined vector computers, and similar tools exist for very small MIMD systems (with 2 or 4 processors), but for large scale MIMD systems the *user* must frequently carefully select, analyse and implement suitable parallel algorithms. On some MIMD systems, some parallelized basic linear algebra algorithms may be available, either from the manufacturer or from software companies, but this was certainly not the case with transputer arrays. Even when such software is already available, the timing models described in this paper should still be useful.

## 1.3 The Local Memory Message Passing MIMD Computer

The parallel algorithms and timing models considered in this paper have been developed for a particular type of Multiple Instruction, Multiple Data (MIMD) computer, namely arrays of INMOS T800 transputers. The algorithms have been implemented in Occam 2 to validate the theoretical analysis.<sup>2</sup> However, the assumptions made regarding the computer are representative of a substantial *class* of parallel computers, namely local memory message passing MIMD systems, so the algorithms and timing analyses are applicable to other computers in this class.

---

<sup>2</sup>Occam is a parallel language based on the work of Hoare on Communicating Sequential Processes (CSP); see [3] for more details. The transputer was designed to very efficiently implement the CSP paradigm.

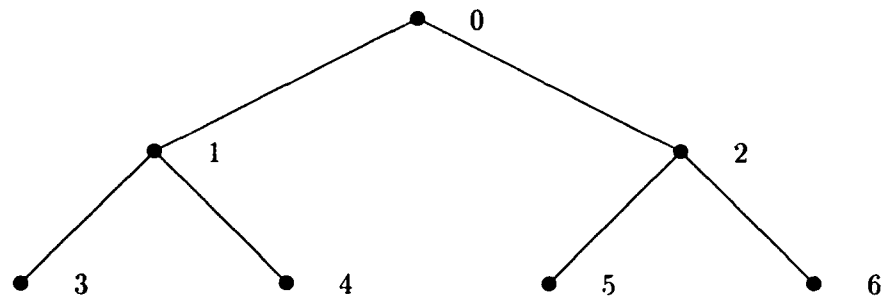


It is important to clearly indicate the properties of this type of computer, so that other researchers with different hardware will be able to establish the suitability of the algorithms, and where modifications to the theoretical analysis will be required, for their computers. The MIMD classification was introduced by Flynn [6] and describes a computer consisting of a number of nodes [7, p.485], each with at least a processing element, which operates independently on its own local instruction stream and data. The further characterization of the machine as *local memory*, *message passing* derives from the memory allocation and communication methods. On a local memory system, all memory is divided up amongst the available processors, and a processor may only directly access its own memory. Access to the memory on other processors is done by explicit message passing, which is *much* slower than direct memory access. The problem of memory contention that complicates the other main competing approach to memory allocation, namely global memory, is removed with this approach, but the absence of global memory can complicate the algorithm — an example will be given later in this paper. It is further assumed that the computer uses explicit message passing over processor to processor communication channels (links) — as opposed to communication over a common bus, for example — for all communication (including both data and synchronization information). It is assumed that each processor has four such links and these links can operate *concurrently* with high efficiency. This theoretical model describes an array of transputers accurately. More details on transputer arrays may be found in [3, 8, 9].

The algorithms derived in this paper use interconnection topologies requiring at most only four links; the number of links required for both the mesh (four) and the binary tree (three) is not a function of the number of processors. These topologies are illustrated in Figures 1 and 2.<sup>3</sup> Four communication links are required to build a two-dimensional grid, a very useful general purpose topology, so four is a reasonable lower bound on the number of links required. The hypercube topology, [3, Section 6.2] and [7], has attracted much attention, and is possibly the most useful general purpose topology currently in use. The hypercube has the attractive property that for a given number of processors, the diameter (the maximum number of links required to connect any two nodes) is smaller than for many other topologies; see [3, Table I]. However, the number of inter-processor links grows as the dimension  $d$  of the hypercube; a hypercube of dimension  $d$  has  $N = 2^d$  processors. While this is a fairly slow (logarithmic —  $\log_2 N$ ) growth in the dimension, and hence number of links required, as a function of the number of processors, this nonetheless imposes limits for systems with a limited number of links. For example, transputer based hypercubes are limited to 16 processors. Fox et. al. have described a number of algorithms that run on hypercubes [7]. Both the topologies (the binary tree and the two-dimensional mesh) used in this paper may be mapped onto hypercube topologies (see [7, Chapter 19] and [7, Chapter 14] respectively), so the algorithms to be described are also suitable for hypercube MIMD computers. It is possible that fully exploiting the greater connectivity of hypercube machines may yield more efficient algorithms than those presented here.

The theoretical results derived in this paper depend on only two machine dependent parameters, viz. the speed of computation and communication. The link concurrency discussed above was exploited to varying degrees, and is discussed in the relevant analysis. The methods developed in this paper permit one to establish at least approximately, from the manufacturer's specifications and benchmarking, whether particular parallel computer hardware will be suitable

<sup>3</sup>The mesh shown in Figure 2 has column wrap-around, but not row wrap-around. The reason for this is rather subtle: a transputer array has to have one link connected to the "host" — typically a PC — and if row-wrap-around was used as well, no spare link would be available. While it is possible to work around this problem, the coding becomes rather messy. Exploiting full wrap-around would reduce the communication cost slightly, but with the pipelined communication used in this paper, the improvement would not be very significant.



**Figure 1:** Interconnection topologies - binary tree dimension 2.

for moment method solutions, and is an important step towards *quantifying* the performance of parallel hardware for important algorithms in computational electromagnetics.

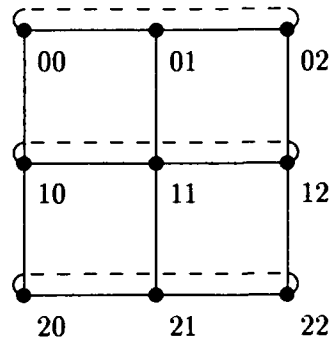
Other researchers [8, 10, 9, 11] have also addressed aspects of parallel processing in electromagnetics, all using transputers, and have shown impressive speed-ups and efficiencies. However, these papers have concentrated on measured results, rendering difficult the application of their results to other types of processor arrays, as well as the extrapolation of the efficiencies of their algorithms to larger arrays. Hafner's paper [8] deals with transputer hardware and software in some detail, as well as the parallelization of a Multiple Multi-Pole program using an early parallel FORTRAN compiler. Nitch's work was on the parallelization of the moment method code NEC2 using a mixture of Occam and FORTRAN. Cramb et al. [9] used the processor farm paradigm for what would be classified as a very "coarse grain" decomposition — essentially the same code was run at different scan angles, with communication only between the controller and the worker processor executing the specific set of scan angles. Russel and Rockway [11] used the ParaSoft EXPRESS operating environment, which provides a number of communications routines of the type implemented explicitly in this paper. Their results for four processors were impressive, but they do not address the scaling behaviour of the algorithm for more processors.

Computer technology moves so rapidly that any paper published giving absolute run-times and computational benchmarks is out of date almost as it goes to print. A comparison of the computational speed obtained with the algorithms described in this paper running on transputer hardware with what may be expected from a typical workstation *at the time of writing* is given in the conclusions of this paper; it must be emphasized that the main thrust of this paper is to describe suitable parallel algorithms for the broad class of local memory MIMD parallel processors — of which the transputer is an contemporary example — and to develop methods for predicting performance of parallel algorithms at least approximately, rather than promoting transputer technology *per se*.

## 2 A Parallel Conjugate Gradient Algorithm

### 2.1 Iterative Algorithms and the Conjugate Gradient Algorithm

Over the past decade, much effort has been expended in the application of iterative methods, and in particular the conjugate gradient and related algorithms, to computational electromagnetics.



**Figure 2:** Interconnection topologies - mesh (lattice) with column wrap-around. See text for further discussion of the wrap-around.

Representative references may be found in Wang's recent book [12, p.68].<sup>4</sup> Golub and Van Cleave's paper provides a recent and comprehensive review of the mathematical history of the algorithm, with an annotated bibliography [14]. A compact description of iterative methods in general and the conjugate gradient algorithm in particular may be found in Jennings [15, Chapter 6]. Regarding parallel iterative algorithms, very little appears to have been published on solvers for full matrices, and what has been published has been frequently directed at different architectures, for example the recent book by Dongarra *et al.* [16] on solving linear systems, which concentrates on vector and shared memory computers.

The CG method, extended for the general case of a matrix  $[A]$  with complex entries where the matrix is not known to be positive definite, is as follows [15, pp.220-221]:

$$\begin{aligned}
 [u_k] &= [A][p_k] && \text{Step 1} \\
 \alpha_k &= \frac{||[\bar{r}_k]||^2}{||[u_k]||^2} && \text{Step 2} \\
 [x_{k+1}] &= [x_k] + \alpha_k[p_k] && \text{Step 3} \\
 [r_{k+1}] &= [r_k] - \alpha_k[u_k] && \text{Step 4} \\
 [\bar{r}_{k+1}] &= [A]^T[r_{k+1}] && \text{Step 5} \\
 \beta_k &= \frac{||[\bar{r}_{k+1}]||^2}{||[\bar{r}_k]||^2} && \text{Step 6} \\
 [p_{k+1}] &= [\bar{r}_{k+1}] + \beta_k[p_k] && \text{Step 7}
 \end{aligned} \tag{3}$$

with initial values  $[r_0] = [b] - [A][x_0]$  and  $[\bar{r}_0] = [p_0] = [A]^T[r_0]$ . This algorithm is suitable for application to the matrix set up by the method of moments. Later in this paper, the question of

<sup>4</sup>There has been a long-running debate in the electromagnetics literature on the relationship of the "direct" application of the CG method to the underlying operator equation as opposed to the use of the method as a matrix solver for the matrix set up by the method of moments, see for example [13] and more recently [5, Chapter 2]. This point will not be pursued further in the present paper, which is directed at the solution of the matrix equation set up by the method of moments.

Step	Complex operations	Real FLOP count
1	$2M^2$	$8M^2$
2	$4M$	$16M$
3	$2M$	$4M$
4	$2M$	$4M$
5	$2M^2$	$8M^2$
6	$4M$	$16M$
7	$2M$	$4M$

**Table 3:** FLOP count of conjugate gradient algorithm.  $M$  is the number of unknowns or equivalently, the dimension of the matrix.

whether the convergence of the CG method justifies its application to a full matrix is discussed.

Note that there are a number of very closely related conjugate gradient algorithms; one recently discussed in the electromagnetics literature is the bi-conjugate gradient (BiCG) method [17]. The author has also implemented the BiCG algorithm; the modifications required to implement it have very little effect on the timing analysis. While the BiCG algorithm sometimes accelerates convergence [18], it can also slow down convergence or stagnate [5, Section 6.8],[17]. Pre-conditioning is also widely used in the Finite Element community to accelerate the convergence of the CG method; unfortunately, previous work by the author on the application of pre-conditioning indicated that it was not suitable for moment method matrices [18]. The reason for this is not clear.

The floating point operation (FLOP) count per iteration is shown in Table 3, retaining only the largest order term for each operation. (Because of this, a term  $-2M$ , with  $M$  the number of unknowns, is missing in the real operations counts in both Steps 1 and 5; this comes from the number of additions, which is actually  $M(M - 1)$ , not  $M^2$ . The impact on the analysis is minimal; it is convenient to use the  $M^2$  approximation for the parallel matrix-vector analysis, and this also indicates clearly the difference between the parallelized matrix-vector products and the unparallelized vector operations in subsequent results.) Note that  $\alpha_k$  and  $\beta_k$  in Steps 3, 4 and 7 are real, not complex, and this affects the conversion from complex to real FLOPs. One complex addition is equivalent to two real FLOPs and one complex multiplication is equivalent to six real FLOPs; since it is the number of additions and multiplications that dominate the FLOP count, and furthermore the addition and multiplication FLOP counts are almost identical, an average factor of four can be used. (On most modern processors, the time required for a floating point addition and a floating point multiplication are approximately the same: benchmarking the transputer yielded *exactly* the same times for both operations.)

## 2.2 Parallel Matrix-Vector Products

From Table 3, the computationally expensive parts of the CG method can be seen to be the two matrix-vectors products — Steps 1 and 5, of  $O(M^2)$  whereas the other steps are of  $O(M)$  — hence efficient parallel matrix-vector product algorithms, taking into account the hardware limitations discussed in the Introduction, are required. (The work of Fox *et al.* discusses parallel matrix-vector products for hypercube architectures [7, Section 21-3.4], and uses a decomposition different from that considered here).

The product of a  $M \times M$  matrix by a vector of length  $M$  can be considered from two viewpoints. The first is as the forming of  $M$  inner products. These inner products can be computed in parallel.

The second approach is as the forming of  $M^2$  products, followed by an accumulation process. The  $M^2$  products can be computed in parallel, and the accumulation process can be parallelized. The computational dependence of both is very similar — detailed expressions will be shown shortly. These viewpoints imply at least the following two possibilities for forming a parallel matrix-vector product:

- *Row-block decomposition*: Splitting up the matrix by row block, *broadcasting* the vector to all processors, performing the inner products in parallel and then *gathering* together the different parts of the vector split up over the processors

*or*

- *Column-block decomposition*: Splitting up the matrix by column, *scattering* the vector over the processing array, performing partial inner products in parallel, and then *accumulating* the resultant vector. This is a special case of the  $M^2$  parallel product approach, with all the elements of a column *clustered* (grouped) on a processor, and entire columns clustered in turn.

The four communications paradigms required by the two different decompositions can be formally defined as follows, assuming  $N$  processors and a matrix dimension of  $M$ :

1. *Broadcast*: This process broadcasts identical copies of the same vector to all the elements of the array.
2. *Gather*: This process builds a vector up from its  $N$  disjoint sections of length  $M/N$  distributed over the array after the parallel matrix/vector product.
3. *Scatter*: This process is the inverse of *gather* in that it scatters a vector over the array so that each of the  $N$  processors has a different vector of length  $M/N$ .
4. *Accumulate*: This process accumulates the partial inner products resulting from the column-block decomposition.

A graphical illustration of the operation of the two possible algorithms may be found in [3] and [5, Chapter 4], where the communication algorithms are also described in more detail.

The next stage of the development of a parallel algorithm is the identification of a suitable topology, i.e. interconnection topology. This issue has been addressed in detail in [3], [5, Chapter 3], and also in [8], and the restrictions imposed by the transputer hardware have already been discussed in the Introduction. Considering the type of communications required, the binary tree, an example of which is shown in Figure 1, is a natural topology for this problem, for the following reasons. It is only necessary to communicate information to and from the (controlling) processor at the top of the tree from and to other lower level processors, and not from one side of the tree to the other. Thus for approximately the same number of processors, the *effective* diameter of the binary tree is actually one less than the diameter of the equivalent hypercube. The processor at the top of the tree can either be used purely for co-ordinating the process, or can also share the workload. The algorithm described here follows the former process. It is possible to use a ternary tree — and the enhanced parallel communications will produce a more efficient algorithm — but this does not map conveniently onto available arrays, where the available number of processors generally follows some power of two. Thus the choice of topology is motivated not only by the algorithm, but also by the available hardware, and typical configurations thereof.

```

begin{broadcast section: worker}
  receive vector from higher processor
  if (not at bottom of tree)
    then
      par
        send vector to lower left processor
        send vector to lower right processor
      end{par}
    else SKIP
  end{if}
end{broadcast section: worker}

```

Figure 3: Pseudo-code for broadcast: worker process

Having identified the parallelism in the problem, the next stage of algorithm analysis is the development of timing equations. These will allow the prediction of the speed-up and efficiency defined in equations (1) and (2). The timing equations have been derived in [3] and [5, Chapter 4] and only the results will be presented. Defining the time required to send one complex word, consisting of the real and imaginary parts — 8 bytes in single precision and 16 bytes in double precision on a transputer, and indeed any system implementing IEEE arithmetic — from one processor to another directly connected processor as  $t_{comm}$ , it may be shown that the communication requirements of the matrix-vector product algorithms for  $M \gg 1$  are as follows [3] and [5, Chapter 4]:

$$t_{broadcast} \approx M d t_{comm} \quad (4)$$

$$t_{gather} = M[1 - d/N] t_{comm} \quad (5)$$

$$t_{scatter} = M[1 - d/N] t_{comm} \quad (6)$$

$$t_{accumulate} = M d t_{comm} \quad (7)$$

where  $d$  is the depth of the binary tree. Since the top-most processor is used purely for coordinating the process, the number of worker processors is  $N = 2^{d+1} - 2$ .

It is important to note that in deriving these results, it has been assumed that the communications parallelism available on a transputer has been exploited — this has been discussed in the Introduction. Figure 3 shows an example of this for the broadcast primitive running on the worker processors. The algorithms are most conveniently documented using pseudo-code — flowcharts are very rarely used for parallel algorithms. The pseudo-code used, loosely based on Pascal, is formally defined in [5, Section 3.7]. The meaning of the code should be intuitively obvious to anyone used to high-level, structured languages. The only construct that may be new is the **par** construct; the code stubs within **par ... par{end}** are executed in parallel.

Note also that there is a certain amount of computation that occurs after each communication phase with the accumulate paradigm, arising from the addition of two vectors at each level; this should be included in the overall compute time. The additional term is  $2Md$  (the factor 2 arising from the conversion from complex to real arithmetic). The use of pipelining, to be discussed later, has not been considered here.

The amount of computation involved in a matrix-vector product is  $M^2$  multiplications and  $M(M-1)$  additions. Thus the total amount of computation is approximately  $2M^2$  complex flops or  $8M^2$  real flops. This is  $T_s$ , the time for the serial operation. The time for the parallel operation,  $T_p(N)$ , is the sum of the computation time for the parallelized matrix-vector product, viz.  $8M^2/N$ , and the communication time for either the row-block or column-block decomposition. The details of the derivation of the speed-up and efficiency have been given in [3]. Only the result for the following important case will be shown. If  $n_r$  is defined as the number of rows per processor,  $n_r = M/N$ , then for  $N \gg 1$ ,  $N \approx 2^{d+1}$ , hence  $d \approx \log_2 N - 1$ , and the following approximate formulae for  $\epsilon$  is obtained:

$$\epsilon = \frac{1}{1 + \frac{\beta}{8n_r} \log_2 N} \quad (8)$$

where  $\beta = t_{comm}/t_{calc}$  is the ratio of communication to computation speed.  $t_{calc}$  is the time required for a real floating point addition or multiplication. This formula is very important; it indicates clearly that the matrix-vector product scales essentially with  $n_r^{-1}$ , the inverse of the number of rows per processor, and rather weakly (logarithmically) with the number of processors,  $N$ . Hence, for a given  $n_r$ , the efficiency is almost independent of the number of processors. This prediction is confirmed by the measured results shown in Figure 7.

In reality, the dimension of the problem will not usually be an integral multiple of the number of processors. This can be handled by either loading different processors differently or by padding the matrix and vector with the necessary zeros. This can be incorporated into the preceding analysis by replacing  $n_r$  by  $\lceil n_r \rceil$ . The effect on a plot of the efficiency as a function of  $M$  (or  $n_r$ ) is to replace the smooth curve implied by equation (8) by a staircase function. This point will be understood as read in the rest of the paper.

The actual run-time of the algorithm can be obtained approximately from  $t_{calc} \frac{8M^2}{S}$ , indicating the obvious importance of maximizing  $S$  for a given  $N$ .

### 2.3 The parallel CG algorithm

The timing analysis for the matrix-vector product of the preceding subsection can now be incorporated into a parallel conjugate gradient algorithm, and  $S$  and  $\epsilon$  predicted. The algorithm exploits the complementary roles of the row- and column-block decomposition; the matrix-vector product is done using the row-block decomposition and the (Hermitian) transpose matrix-vector product is done using the column-block decomposition (with the necessary change of sign of the imaginary part of the matrix entries). This avoids having to either explicitly form the matrix transpose during each operation — a very expensive operation on a parallel processor with *local* memory, since this requires an  $O(M^2)$  interchange operation at each iteration — or store an additional copy of the Hermitian transpose of the matrix — and thus double the memory requirements of the code. This important contribution was the author's [3], and has not been published elsewhere, to the best of his knowledge. It is notable that an operation as simple as forming the transpose of a matrix — a trivial interchange of indices on a serial processor — can pose a major problem on a parallel system. Pseudo-code for the algorithm is given in Figures 4 and 5 for the master and workers respectively. Only the matrix-vector products have been parallelized (Steps 1 and 5 in Table 3); the other vector update operations are performed on the master processor at the top of the tree.

From Table 3, the serial time is

$$T_s = (16M^2 + 44M)t_{calc} \quad (9)$$

```

process[master.cg]
begin
  initialization
  while (not finished)
    begin
      broadcast p.k
      gather u.k
      compute alpha.k
      update x.k+1 and r.k+1
      scatter r.k+1
      accumulate r.bar.k+1
      compute beta.k
      update p.k+1
      compute and print normalized residual
      check termination
    end
  end{while}
end{process[master.cg]}

```

Figure 4: Pseudo-code for parallel CG algorithm: master process

```

process[worker.cg]
begin
  initialization
  while (not finished)
    begin
      broadcast p.k
      perform matrix-vector product
      gather u.k
      scatter r.k+1
      perform transpose matrix-vector product
      accumulate r.bar.k+1
      check termination
    end
  end{while}
end{process[worker.cg]}

```

Figure 5: Pseudo-code for parallel CG algorithm: worker process



<i>Precision</i>	<i>Operation</i>	<i>MFLOP/s</i>
Single	Addition	0.53
Double	Addition	0.38
Single	Multiplication	0.53
Double	Multiplication	0.38

**Table 4:** Computation benchmarks on the University of Stellenbosch's T800 transputer array.

The parallel time is the sum of the parallelized matrix-vector products, the unparallelized vector operations and the additional computational overhead of the accumulate paradigm, and the communication requirements of the broadcast, gather, scatter and accumulate paradigms:

$$T_p = (16M^2/N + 44M + 2dM)t_{calc} + (2M[1 - d/N] + 2Md)t_{comm} \quad (10)$$

Forming the quotient of  $T_s$  and  $T_p$  and simplifying yields

$$\epsilon \approx \frac{1 + \frac{2.75}{M}}{1 + \frac{N}{M}(2.75 + 0.125d + \frac{[1+d(1-\frac{1}{N})]\beta}{8})} \quad (11)$$

Note that this result is actually the efficiency of one iteration; since by far the majority of time required by the algorithm is in the iterative cycles, the algorithm as a whole can be characterized by its performance per iteration.

Under the assumption  $M, N \gg 1$ , this can be simplified to

$$\epsilon \approx \frac{1}{1 + \frac{N}{M}(2.75 + 0.125d + \frac{\log_2 N \beta}{8})} \quad (12)$$

Attention must be paid to correctly terminating parallel algorithms: if not done correctly, certain processes will never terminate, and re-initialization of the array may be required before any other code will load. In the code developed by the author, the termination criteria is that either the normalized residual error must have decreased to less than the user-specified value or that some maximum number of iterations must have been executed (the conventional criteria for an iterative algorithm). The former can only be determined by the master processor. Hence it is necessary for the master process, at the end of each iteration, to monitor the termination criteria. If one (or both) of the termination criteria has been satisfied, then the master must explicitly inform the workers, who then inform the lower level workers and terminate their execution. The configuration program that loads the worker processors and correctly allocates software abstractions (channels in Occam) to hardware (links on a transputer) for an arbitrary depth of binary tree also requires attention; this is dependent on the specific language and configuration meta-language. A suitable configuration for the Occam code developed by the author is given in [5, Appendix A].

This analysis requires two fundamental parameters to characterize the machine: the computation and communication speeds. The most reliable way of obtaining this data is by *benchmarking* - actually measuring the performance of the system under conditions simulating those of the actual code. Two simple benchmarks were developed: the first tested computation speed and the second communication speed. Such benchmarking is necessary for any parallel computer; pseudo-code useful for benchmarking local memory MIMD systems is presented in [5, Section 4.7]. Results are shown in Tables 4 and 5. The parameter  $\beta$  can now be computed from the benchmark results

<i>Precision</i>	<i>MByte/s</i>
Single	1.32
Double	1.39

**Table 5:** Communication benchmarks on the University of Stellenbosch's T800 transputer array.

<i>Precision</i>	$\beta$
Single	3.22
Double	4.37

**Table 6:**  $\beta = t_{comm}/t_{calc}$

for the case of single and double precision. The numerical values given in Table 6 are for the transputer array used to generate the results that follow.<sup>5</sup>

## 2.4 Results and Discussion

This section describes results obtained by the author using his Occam 2 implementation of the algorithm. It was run on a 64 transputer array, developed in South Africa by the Council for Scientific and Industrial Research. The array is known as the Massively Concurrent Computer/64 (MC<sup>2</sup>/64, or MC<sup>2</sup> in this paper).<sup>6</sup> The array has been described in [3, Section 4.1]. At the time of running the timing tests, it was only possible to use half the array, for technical reasons: firstly, the memory was not homogeneously distributed, and secondly, some problems with the inter-cluster switching (from one "cluster" of 16 processors to another) were being experienced. These results represent the experimental validation of the timing models developed. Although the pseudo-code given in Figures 4 and 5 appears simple, much detail — especially in the communication routines — is hidden, and the debugging was very time-consuming and tedious, due to the absence of interactive parallel debuggers. This code was also developed before useful books on the subject such as [7] were available.

Measured efficiencies are shown in Figure 6. Theoretically, equation (11) predicts that the efficiency should be a function mainly of the number of rows per processor,  $\frac{M}{N}$ , and a weak function of  $d$ , the depth of the tree. These predictions are confirmed in Figure 7. *Thus this parallel CG algorithm exhibits a most desirable property - it scales with the number of rows per processor. With a given number of rows per processor, the efficiency of the algorithm is a rather weak function of the number of processors.* The measured and predicted results for 30 workers are shown in Figure 8. The maximum problem size is limited by the available memory; at the time of writing a maximum of 64MB of usable memory was available.

It will be noted that in Figure 8, the measured and predicted curves agree very well regarding the *shape* of the curve, but there is an offset between the measured and predicted curves. Similar results — not shown in this paper — were obtained for other numbers of processors. It should be

<sup>5</sup>One of the reviewers queried these benchmarks. However, these results agree closely with those reported in [8, Table I] for the FORTRAN benchmarks, when the off-chip RAM is used. Using on-chip RAM yields rather faster results [8], but there is only 4kB of this, so any real application program has to use the off-chip RAM. The computational benchmark was constructed to avoid measuring loop overhead.

<sup>6</sup>The name "Massively" seems rather presumptuous in retrospect, but when initially mooted, the system was indeed massive.

noted that the aim of the modelling is not to be able to predict the performance exactly in the sense that one predicts an antenna's radiation pattern; the aim is simply to indicate trends and determine whether the performance (efficiency) will be satisfactory for the problems of interest. Furthermore, the predictions serve as a check on the correct functioning of the code. Possible causes of the differences are latency (the time to initiate communication), loop overhead and differences between the coding and the model caused by some usage restrictions in Occam. More details may be found in [5, Section 4.8].

The measured data shown was obtained from *PARNEC*, a parallel version of the thin-wire part of the moment method code NEC2 developed by the author in Occam 2 [5, Chapter 6]. Double precision was used.

The efficiency of the parallel CG algorithm that has been described in this paper could be further increased by exploiting communication pipelining, a concept that will be described with reference to the parallel LU algorithm. Some further comments on improving the efficiency of the algorithm will be made in the Conclusions of this paper.

### 3 A Parallel LU Algorithm

#### 3.1 The Basic LU Algorithm

The LU method is probably the most widely used algorithm for the solution of square systems of linear equations. Given a system with a moderate number of equations, it is usually the best algorithm to use, provided that the system is not extraordinarily ill-conditioned. On serial processors<sup>7</sup>, LU decomposition followed by forward and backward substitution is always better to use when solving a system of equations than forming the explicit inverse of the matrix [19, p.347]. Given the fundamental role of the LU algorithm, the development of an *efficient* algorithm suitable for a local memory MIMD array is an essential research topic for parallel computational electromagnetics.

Before considering the parallel version of the LU algorithm, it is necessary to review briefly the serial form of the algorithm. The LU algorithm factors a matrix *A* into the product of an upper (*U*) and lower (*L*) triangular matrix. The diagonal elements of *L* are most commonly chosen as 1 — although other choices are also useful, for example Choleski decomposition.<sup>8</sup> The algorithm can be found in virtually any book on matrix algebra, for example [19, p. 359]. The algorithm consists of *M* main steps. Step *i*, which computes the *i*-th row of *U* and the *i*-th column of *L*, is repeated for *i* = 1, ..., *M* - 2 and is defined as follows:

**begin{Step *i*}**

$$u_{i,i} = \frac{1}{l_{i,i}}[a_{i,i} - \sum_{k=0}^{i-1} l_{i,k} u_{k,i}]$$

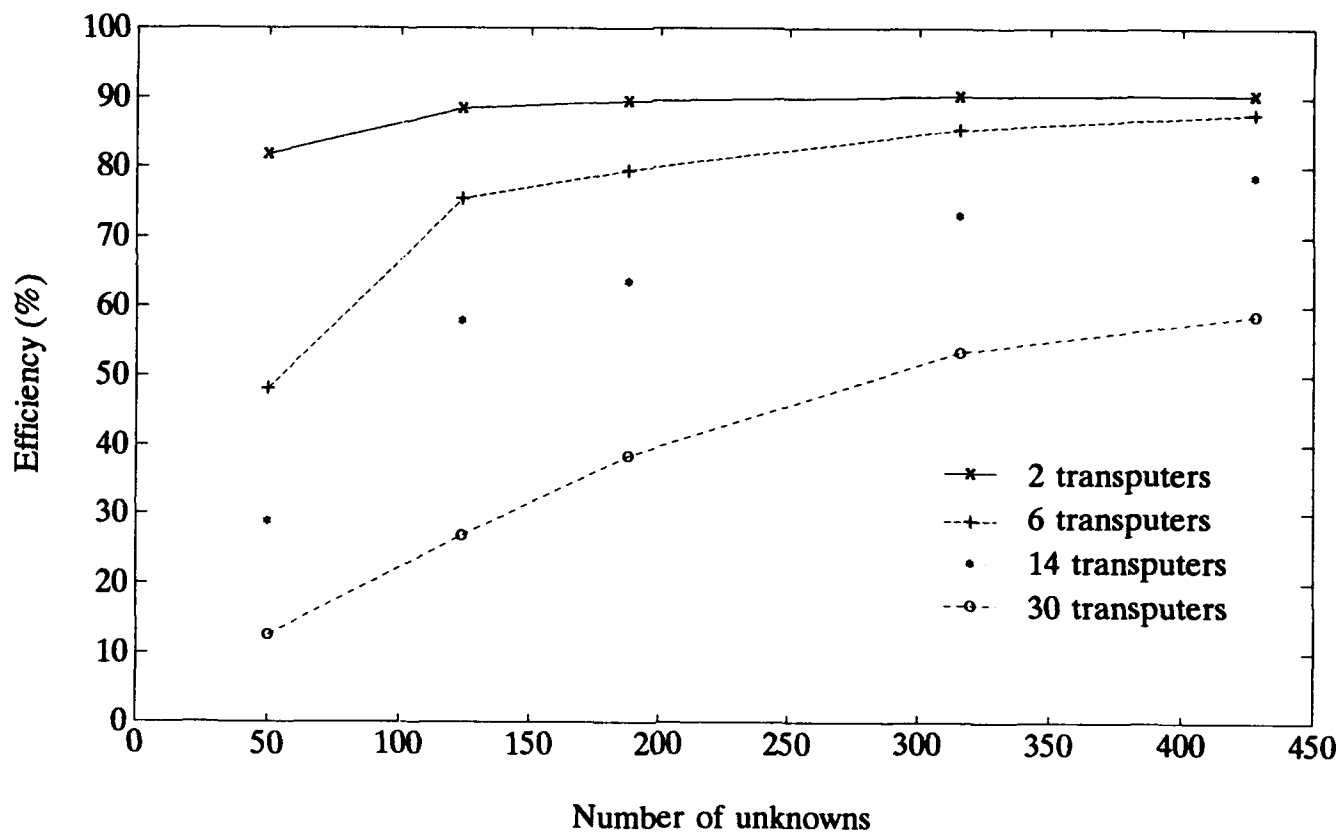
Repeat for all *j* = *i* + 1, ..., *M* - 1:

$$u_{i,j} = \frac{1}{l_{i,i}}[a_{i,j} - \sum_{k=0}^{i-1} l_{i,k} u_{k,j}] \quad (13)$$

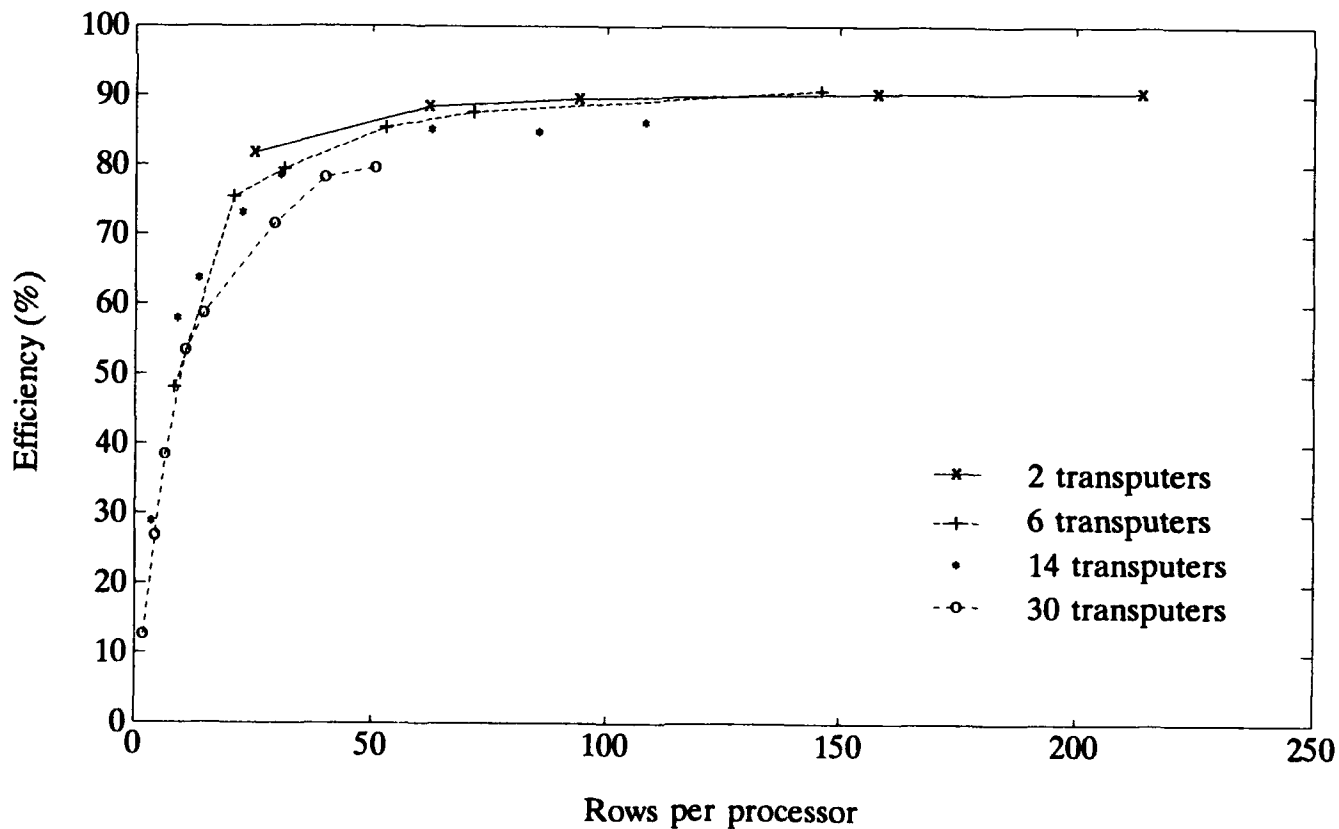
$$l_{j,i} = \frac{1}{u_{i,i}}[a_{j,i} - \sum_{k=0}^{i-1} l_{j,k} u_{k,i}] \quad (14)$$

<sup>7</sup>It was brought to my attention by a reviewer that some researchers have concluded that this may not be true on certain parallel systems such as the Connection Machine.

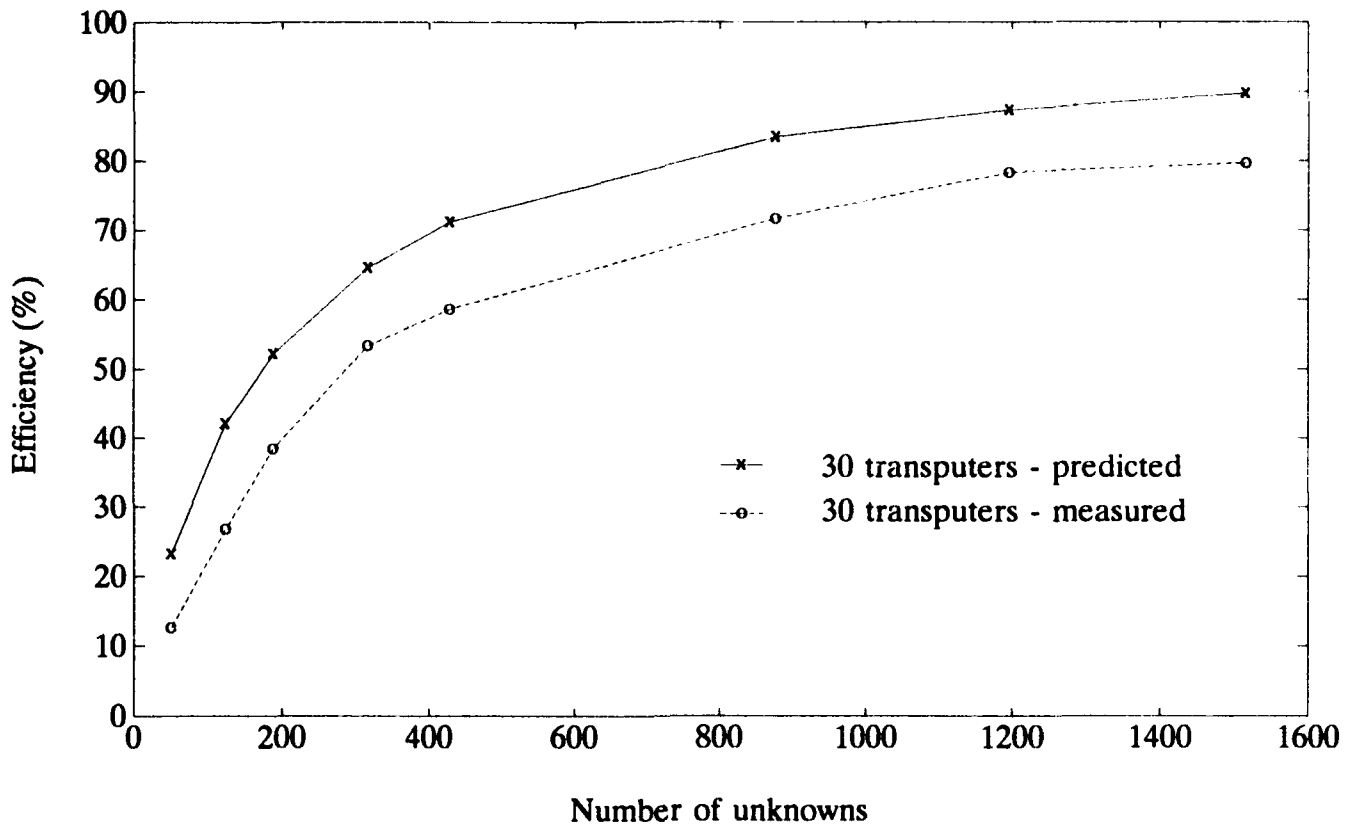
<sup>8</sup>Note that Choleski decomposition is only applicable to symmetric positive definite matrices [15, p.107]. Matrices set up by the moment method do not generally have these properties.



**Figure 6:** Measured efficiencies of the double precision parallel conjugate algorithm versus unknowns for the MC<sup>2</sup>.



**Figure 7:** Measured efficiencies of the double precision parallel conjugate algorithm versus rows per processor for the MC<sup>2</sup>.



**Figure 8:** Measured and predicted efficiencies of the double precision parallel conjugate algorithm (30 worker transputers) for the MC<sup>2</sup>.

**end{Step i}**

$a_{i,j}$ ,  $l_{i,j}$  and  $u_{i,j}$  represent the  $i,j$ -th element of the  $[A]$ ,  $[L]$  and  $[U]$  matrices respectively;  $i, j \in \{0, 1, \dots, M-1\}$ ;  $M$  is the dimension of the matrix. The matrix entries have been numbered from 0 to  $M-1$  for coding convenience: an array  $a$  in Occam is numbered  $a_0, a_1, \dots$ .

The algorithm requires special treatment for Step 0 and Step  $M-1$  [19, p.359], and if at any stage  $l_{i,i}u_{i,i} = 0$ , the algorithm is terminated with an error message to the effect that factorization is impossible. Provided that the matrix is not singular, *pivoting* may be used in such cases — and is advisable whenever the matrix is not known to be well conditioned. Pivoting is a strategy to optimize numerical stability by ensuring that the largest (in some sense) element is on the diagonal. *Maximal column* or *partial pivoting* and *maximal* pivoting [19, p.330–3] are two well-known algorithms; the former involves searching the column below the diagonal, the latter the entire active region, to use the nomenclature of this paper. Bisseling and van de Vorst [20] have shown that partial pivoting may be incorporated into the parallel LU algorithm implemented in this paper without a major effect on the efficiency of the algorithm; the effect on equation (22) is to replace the  $\frac{1}{2}$  by  $\frac{3}{2}$ . However, the coding becomes substantially more complicated than that already required and is not at present incorporated into the author's parallel code.

Following the factorization of  $[A]$  into the product of  $[L]$  and  $[U]$ , the unknown left-hand side is solved for in a two-step process; the first step is forward substitution. Consider  $[A][x] = [b]$ , with  $A$  factored as  $[A] = [L][U]$ . Define  $[U][x] = [z]$ . Now the system  $[L][z] = [b]$  can be solved for using forward substitution, since  $[L]$  is lower triangular. Then  $[x]$  can be solved using backward substitution from  $[U][x] = [z]$  since  $[U]$  is upper triangular.

It may be shown that the timing requirements of LU decomposition are approximately  $\frac{M^3}{3} + O(M^2) + O(M)$  additions and approximately the same number of multiplications; see [15, p.109]. The constants associated with the lower order terms are small integers, so for all practical purposes, the amount of work required is  $\frac{2M^3}{3}$  operations. The factor of 2 comes from the additions *and* multiplications. Similarly, the dominant term in the time for forward substitution is  $M^2$  operations, and the same result also holds for backward substitution.

### 3.2 Previous Work on Parallel LU Algorithms

This discussion of the serial algorithm now leads to the question of the identification of the parallelism in the algorithm. Compared to the CG algorithm discussed in the preceding section, the parallelism is hardly obvious. Nonetheless, very efficient parallel algorithms can be developed.

Since LU decomposition is such a fundamental algorithm in linear algebra, much work has been done, but very often the work is not applicable to the problem of a full matrix, without any special properties. Brief reviews of parallel LU decomposition may be found in [21, 22]; a rather more recent review paper is [23]. The present paper is based on recent work by van de Vorst and Bisseling [24, 20]; the algorithm used is essentially identical to that of Fox *et al.* [7, Chapter 20], although the very different approaches used to present their algorithms by Bisseling and van de Vorst on the one hand, and Fox *et al.* on the other, make this similarity initially obscure. Van de Vorst's work [24] is particularly difficult to read — cryptic is not an exaggeration for someone unfamiliar with the use of formal methods in computer science — and Fox's work, while far easier to read, is for a banded matrix, hence the difficulties in recognizing the similarity.

### 3.3 A Parallel LU Algorithm - a Graphical Description

The essence of the parallel algorithm is the following observation. Instead of waiting for Step  $i$  to compute  $u_{i,j}$  and  $l_{j,i}$ , as in the serial algorithm described in the previous section, the summations in equations (13) and (14) may be performed as soon as data is available, given sufficient processors ( $N = M^2$ ). As an example, the first summation for each element of row 1 of  $U$  may begin as soon as the relevant element of row 0 of  $U$  and column 0 of  $L$  is available. All the summations required for row 1 may of course be performed in *parallel*, since there is no dependence *within* a row of  $[U]$  or a column of  $[L]$  (other than on the diagonal element for the final division). Similarly, the first summations for row 2, 3 etc. may also begin as soon as the results of row 0 and column 0 are available. One could of course perform the serial algorithm in exactly the same way, but in the serial case, nothing would be gained, and the algorithm would appear unnecessarily complex. The required summations for row  $i$  of  $U$  and column  $i$  of  $L$  are thus computed using a series of *partial sums* performed in *parallel* at each step which *terminates* in Step  $i$ . Hence the maximum degree of parallelism in this algorithm is  $M^2$ . As will be noted shortly, the algorithm requires at least  $2M$  steps to execute. A more detailed explanation may be found in [4], which discusses in a tutorial fashion the mode of thinking required to identify the parallelism inherent in the algorithm.

A parallel algorithm implementing the above is given in pseudo-code in Figure 9. This algorithm assumes the diagonal elements of  $[L]$  to be 1. Note that the pseudo-code assumes  $M^2$  processors; if this is not the case, then *clustering* is required. It should be appreciated that efficiently implementing the clustering and communications made the actual Occam code much more complex than the pseudo-code shown. A matrix  $[X]$  is used in the pseudo-code; when the algorithm terminates the upper triangular part of  $[X]$  is  $[U]$ , and the lower triangular part of  $[X]$  — excluding the diagonal elements, which are 1 by initial choice — is  $[L]$ . If the matrix  $[A]$  is not needed after factorization, then as the computation of elements of  $[X]$  is completed the corresponding elements of  $[A]$  may be overwritten.

```

process[s,t] :
begin
  x[s,t] := a[s,t] {initialize matrix}
  k := 0 {initialize global clock}
  while k < n do
    begin
      if k < min(s,t) then
        begin{active}
          par
            receive l[s,k] from process [s,k]
            receive u[k,t] from process [k,t]
          end{par}
          x[s,t] := x[s,t] - l[s,k]*u[k,t]
        end{active}
      else if k = t AND s > t then
        begin{critical}
          receive u[k,k]
          x[s,t] := x[s,t] / u[k,k] {note k=t in this case!}
          send x[s,t] to all processes[s,q] with q > k
        end{critical}
      else if k = s AND s < or = t then
        begin{pseudo-critical}
          send x[s,t] to all processes[q,t] with q > k
        end{pseudo-critical}
      else if k > min(s,t) then
        SKIP {passive}
      k := k + 1
    end
  end{while}
end. { process[s,t] }

```

Figure 9: Pseudo-code for the parallel LU algorithm; adapted from [24].

The algorithm can be most easily understood graphically. Figures 10 to 12 show the evolution of the algorithm for a matrix of dimension 3 on a 3 by 3 array of processors, i.e. one processor per element, the upper limit of the parallelism that can be extracted with this algorithm. The • and \* represent elements that are critical i.e. in the last stage of computation. (The \* represent the row of  $[U]$  in the final stage of computation. The choice of the diagonal elements of  $[L]$  as 1 means that no computation is required, but the values must still be communicated, hence the distinction and the name pseudo-critical, used in the pseudo-code). The o represents elements that are active, i.e. forming the partial sums. Blank entries represent passive elements, where no work is performed, since the relevant element of  $L$  or  $U$  has been computed in a previous step. The echelons of completed elements step diagonally downwards in an almost wave-front fashion.

$$\begin{bmatrix} * & * & * \\ \bullet & o & o \\ \bullet & o & o \end{bmatrix}$$

**Figure 10:** Step 0 of LU decomposition

$$\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & * & * \\ \cdot & \bullet & o \end{bmatrix}$$

**Figure 11:** Step 1 of LU decomposition

$$\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & * \end{bmatrix}$$

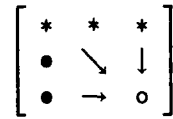
**Figure 12:** Step 2 of LU decomposition

- passive elements
- critical elements
- \* pseudo-critical elements
- o active elements

It is useful to give an example describing how the algorithm given in Figure 9 and illustrated in Figures 10 to 12 proceeds. It is assumed for this discussion that there are  $M^2$  processors, i.e. one processor per matrix element. The initialization of  $[X] = [A]$  establishes the first row of  $[U]$  — actually before the algorithm has started. (This is because of the choice of diagonal  $[L]$  elements).

- On step 0, the first column (column 0) of  $[L]$  is computed, and then this column, as well as the first row (row 0) of  $[U]$  is sent to all the critical processes so that the partial sums can be





**Figure 13:** Communication in parallel LU algorithm, Step 0. The arrow symbols are defined in the text.

computed. Note that by the end of step  $k = 0$ , the computations for the second row (row 1) of  $[U]$  have been completed.

- On step 1, the second column (column 1) of  $[L]$  is computed, and this column, as well as the second column of  $[U]$ , can be sent to all remaining critical processes so that ongoing partial sums can be computed. By the end of step  $k = 1$ , the computations for the third row of  $[U]$  (row 2) have been completed.
- The algorithm proceeds thus, until  $k = M$ . (3 in this case).

Note that even given  $M^2$  processors, at each step  $i$ , corresponding to one of the Figures 10 to 12, the algorithm given in Figure 9 needs two discrete computational “ticks”: firstly, to compute the  $i$ -th column of  $L$  — in parallel — and secondly, to then update the partial sums on the active processors— also in parallel. Hence with  $M$  processors the algorithm will take  $2Mt_{calc}$  to terminate, assuming the times for floating point addition, subtraction, multiplication and division to be similar.

Figure 13 shows the communications executed by the algorithm in Step 0. In this figure, the  $\downarrow$  indicates communication to all the active elements of the column, and similarly the  $\rightarrow$  indicates communication to all the active elements of the row. The  $\searrow$  symbol indicates both  $\downarrow$  and  $\rightarrow$ .

Note how fine the grain of parallelism is compared to some other published applications; see for example Cramb *et al.* [9]. They use a processor farm application for antenna array modelling, and decompose their problem by scan angle, producing a parallel system requiring very little data interchange: essentially data initialization, then collection of the finished computations. Such an application is rather easier than those considered in this paper, since far less attention need be given to highly efficient coding.

### 3.4 Topology, Clustering, Load Balancing and Communications

Given enough processors, the natural topology in the case of LU decomposition is a two-dimensional mesh, reflecting the two-dimensional matrix. The row and column communication shown in Figure 13 can also be implemented very efficiently on such a mesh. However, as with the CG algorithm, the problems of interest are large-grained, where many unknowns must be grouped (or clustered) on each processor. A new problem, not present in the CG algorithm, emerges with the LU algorithm, viz. load balancing. Inspection of Figures 10 to 12 show the problem; the work in each row and column decreases as the algorithm proceeds, resulting in idle processors, producing a lower bound on the efficiency of only approximately 33% [5, Section 5.8]. Hence the topology required for an efficient LU algorithm must not only minimize the communication cost, but also provide a solution to the load balancing problem. The solution to the latter is clearly to interleave rows or columns in some fashion so that the work on each processor remains fairly constant, but this is also clearly linked to the communication cost. Prior to van de Vorst’s work, most LU decomposition algorithms clustered the unknowns either by row or by column. However,

$a_{00}$	$a_{03}$	$a_{06}$	$a_{01}$	$a_{04}$	$a_{07}$	$a_{02}$	$a_{05}$	$a_{08}$
$a_{30}$	$a_{33}$	$a_{36}$	$a_{31}$	$a_{34}$	$a_{37}$	$a_{32}$	$a_{35}$	$a_{38}$
$a_{60}$	$a_{63}$	$a_{66}$	$a_{61}$	$a_{64}$	$a_{67}$	$a_{62}$	$a_{65}$	$a_{68}$
$a_{10}$	$a_{13}$	$a_{16}$	$a_{11}$	$a_{14}$	$a_{17}$	$a_{12}$	$a_{15}$	$a_{18}$
$a_{40}$	$a_{43}$	$a_{46}$	$a_{41}$	$a_{44}$	$a_{47}$	$a_{42}$	$a_{45}$	$a_{48}$
$a_{70}$	$a_{73}$	$a_{76}$	$a_{71}$	$a_{74}$	$a_{77}$	$a_{72}$	$a_{75}$	$a_{78}$
$a_{20}$	$a_{23}$	$a_{26}$	$a_{21}$	$a_{24}$	$a_{27}$	$a_{22}$	$a_{25}$	$a_{28}$
$a_{50}$	$a_{53}$	$a_{56}$	$a_{51}$	$a_{54}$	$a_{57}$	$a_{52}$	$a_{55}$	$a_{58}$
$a_{80}$	$a_{83}$	$a_{86}$	$a_{81}$	$a_{84}$	$a_{87}$	$a_{82}$	$a_{85}$	$a_{88}$

**Figure 14:** Scattered grid distribution; 3 by 3 processor array (mesh). The elements in the upper left corner map onto processor 00 of Figure 2, those in the upper centre onto 01, those in the left centre onto 10 etc.

a better method is to combine these. This double interleaved distribution is also used by Fox *et al.* [7, Section 20-3] for the parallel LU decomposition of a banded LU matrix — the bandedness of the matrix affects the timing analysis, but not the basic algorithm. Fox *et al.* use the term “scattered square decomposition”. It would appear that Fox has priority on the double interleaved distribution, but his early work appeared as internal Caltech reports and van de Vorst’s and Fox’s work appeared in the published literature at much the same time. Van de Vorst’s earlier work appears to have been carried out independently of Fox’s [24], but Bisseling and van de Vorst later acknowledge the similarity [20].

From the viewpoint of minimizing the communication count, van de Vorst [24] has shown that a square mesh distribution is the optimal  $N_1 \times N_2$  grid topology. (Note that a row or column distribution are extreme cases of this general case, in the former case with  $N_1 = N$ ,  $N_2 = 1$ , and vice versa for the latter). This can be confirmed intuitively: with a column or row distribution, the amount of data to be communicated at each step is  $O(M)$  — an entire column (or row) must be communicated — whereas using the grid distribution the amount of data at each step is  $O(\frac{M}{\sqrt{N}})$ . Furthermore, with the grid distribution, the column and row broadcast pipelines can be run concurrently. This will be explained shortly.

With this grid decomposition required to minimize the communication cost, the load balancing problem may be solved very elegantly using a double-interleaved clustering scheme for data distribution [24, 20], whereby *both* row *and* columns are scattered modulo  $\sqrt{N}$  over a square array of  $\sqrt{N}$  by  $\sqrt{N}$  transputers, with  $\sqrt{N} \ll M$ . The distribution of a matrix of dimension 9 on a 3 by 3 array using this double interleaved distribution is shown in Figure 14 for the processor mesh shown in Figure 2. The “wave-front” suggested by Figures 10 to 12 now sweeps cyclically through the processor array, each cycle completing  $\sqrt{N}$  rows and columns. The algorithm terminates after  $M/\sqrt{N}$  cycles. It may be seen by inspection that all the processors remain occupied until the very last cycle of the algorithm. The load-balancing problem is thus alleviated.

In the case where  $M$  is not an integral multiple of  $\sqrt{N}$ , special care is required; the work is divided up as evenly as possible but the processors with one less row and column to work on must be thus explicitly programmed. The method used in the CG code of padding the matrix with rows and columns of zeros is not applicable in this case, since the LU algorithm fails when a zero is encountered on the diagonal.

Formally, the double interleaved distribution is the Cartesian product  $G$  of sets  $G_i \times H_j$  :

$$G \equiv \{G_i \times H_j : 0 \leq i, j < \sqrt{N}\} \quad (15)$$

with

$$G_i \equiv \{s : s \in V \wedge s \bmod \sqrt{N} = i\} \forall 0 \leq i < \sqrt{N} \quad (16)$$

$$H_j \equiv \{t : t \in V \wedge t \bmod \sqrt{N} = j\} \forall 0 \leq j < \sqrt{N} \quad (17)$$

and

$$V \equiv \{s : 0 \leq s < M\} \quad (18)$$

The indices  $i$  and  $j$  refer to processor indices and the indices  $s$  and  $t$  to matrix element indices. As an example, for the 9 by 9 matrix distributed on the 3 by 3 processor mesh shown in illustrated in Figures 2,  $V = \{0, 1, 2, \dots, 8\}$ , and  $G_0, G_1$  and  $G_2$  (and  $H_0, H_1$  and  $H_2$ ) are  $\{0, 3, 6\}$ ,  $\{1, 4, 7\}$  and  $\{2, 5, 8\}$  respectively. The Cartesian product  $G_0 \times H_0$  gives the indices of the 9 elements clustered on processor<sub>00</sub>. The full distribution  $G$  is shown in Figure 14.

An upper bound on the load-balancing complexity can be established as follows. The maximum load is carried by processor <sub>$\sqrt{N-1}\sqrt{N-1}$</sub>  (the processor at the lower right of the processor array). As already discussed, the scattered grid distribution results in a cyclic "sweep" through the processor grid, with  $\sqrt{N}$  Steps per cycle and  $M/\sqrt{N}$  cycles in total. The amount of work in the last cycle — where there is only one element left to update — is approximately  $2(\sqrt{N})$  (the factor 2 comes from the multiplication followed by subtraction, and the  $\sqrt{N}$  from the number of Steps in the cycle); on the preceding cycle  $2(4\sqrt{N})$ ; and so on back to the first cycle with  $2[M/\sqrt{N}]^2\sqrt{N}$ . Summing over all  $M/\sqrt{N}$  cycles yields an upper bound of

$$\frac{2}{3} \frac{M^3}{N} + \frac{M^2}{\sqrt{N}} \quad (19)$$

The first term is clearly the parallelized computations; thus the second term is the additional computational overhead caused by the load-balancing term.

The communications use pipelined, concurrent, row and column broadcast. The pipelines are implemented in software; the concept is to overlap the incoming and outgoing vector to further exploit the parallel link operation possible on a transputer. An example is shown in Figure 15 for one of the communication primitives exploiting pipelining. The effect is to speed up the communications by a factor of almost  $2\sqrt{N}$ , where the factor of 2 derives from the concurrent row and column operation and the  $\sqrt{N}$  from the pipelining. Details and more complete pseudo-code may be found in [5, Section 5.10], and also in [4].

An upper bound for the communication count can be derived by considering the *processor* column carrying the heaviest communication load, namely the right-most column. For the first cycle, the amount of data to be communicated is approximately  $\frac{M}{\sqrt{N}}$  for each Step in the cycle. For the next cycle, the amount of data is  $\frac{M}{\sqrt{N}} - 1$  per Step, and so on. Summing over all the  $M/\sqrt{N}$  cycles yields

$$t_{mesh} \leq \{[(\frac{M}{\sqrt{N}})\sqrt{N}] + [(\frac{M}{\sqrt{N}} - 1)\sqrt{N}] + \dots + [(1)\sqrt{N}]\} t_{comm} \quad (20)$$

There are  $\frac{M}{\sqrt{N}}$  square-bracketed terms in total in the above equation (i.e. the number of cycles), which can be re-written as

$$t_{mesh} \leq \{ \frac{M^2}{\sqrt{N}} - \sqrt{N} \sum_{k=0}^{\frac{M}{\sqrt{N}}-1} k \} t_{comm} \quad (21)$$

and thus

$$t_{mesh} \leq \frac{1}{2} \frac{M^2}{\sqrt{N}} + O(M) \quad (22)$$

```

procedure broadcast_column_to_right(length)
begin
  {initialize pipeline}
  {note: length of vector passed as argument}
  receive vector[1] from left processor
  repeat for i = 2 to length
    par{run pipeline}
      receive vector[i] from left processor
      send vector[i-1] to right processor
    end{par}
  end{repeat}
  {flush pipeline}
  send vector[length] to right processor
end{procedure broadcast_column_to_right}

```

**Figure 15:** Pseudo-code for rightwards pipelined column broadcast procedure. This runs in parallel with similar leftwards column broadcast and upwards and downwards row broadcast procedures.

Bisseling and van de Vorst's result [20, equation (3.19)] has an identical dependence on  $\frac{M^2}{\sqrt{N}}$ , once the necessary change of notation is made.

A theoretical model for the efficiency may now be derived. The serial time, using a conversion factor from complex to real flops of 4 as before, is  $(\frac{8}{3}M^3)t_{calc}$ ; the parallel time is the sum of the computation count, equation (19), and the communication count, equation (22). Summing the last three, using equation (1) and simplifying yields

$$\epsilon = \frac{1}{1 + n^{-1}(\frac{3}{2} + \frac{3\beta}{16\gamma})} \quad (23)$$

where  $n = M/\sqrt{N}$  is the *grain* of the problem, i.e. the number of unknowns per processor, and  $\beta$  has the previously defined meaning.

It is instructive to compare this result with that for the CG solver, re-written using the same notation:

$$\epsilon = \frac{1}{1 + n^{-1}\sqrt{N}(2.75 + 0.125d + \frac{\log_2 N \beta}{8})} \quad (24)$$

Note that the terms in  $n^{-1}$  in the denominator of the respective equations have similar constant multipliers, but in addition the CG equation has a  $\sqrt{N}$  and also a  $\log_2 N$  term. Hence it can be expected that for similar  $n$  that the LU algorithm is more efficient, a result that is confirmed experimentally. *This indicates that a parallel LU algorithm based on a mesh topology scales better than a parallel CG based on a binary tree* — the mesh and binary tree being considered as the “natural” topologies for the CG and LU algorithms respectively, for the reasons already discussed in this paper. To summarize: the CG algorithm scales with the reciprocal of the number of *rows* per processor, whereas the LU algorithm scales with the reciprocal of the square root of the number of *unknowns* per processor, and the latter is the smaller multiplier. This is a most interesting result, considering how initially unsuitable for parallelism the LU algorithm appeared, and is confirmed by the results in Section 3.6.

```

process[s] :
begin
  z[s] := b[s]; k = 0 {initialize}
  while k < n do
    begin
      if k < s then
        receive z[k] from process [k]
        z[s] := z[s] - L[s,k] z[k]
      else if k = s then
        z[s] := z[s] / L[s,s]
        send z[s] to all processes q with q > k
      else if k > s then
        SKIP
      k := k+1
    end
  end. { process[s] }

```

Figure 16: Forward substitution pseudo-code; solve  $[L][z]=[b]$ .

### 3.5 Parallel Forward and Backward Substitution

Following the factorization of  $[A]$  into the product of  $[L]$  and  $[U]$ , the unknown left hand side is solved for using the two-step forward and backward substitution processes already discussed. A parallel version of the forward and backward substitution algorithms is also necessary, not because of the computation time, which is  $O(M^2)$ , but because it is most undesirable to communicate all the elements of the  $[L]$  and  $[U]$  matrices back to a master processor, since the master must then have enough memory to store the entire matrix and the communication procedure takes time. The former is the more serious problem for a typical MIMD array with local memory; sufficient memory is not available on any one node (processor plus memory) to store the entire matrix. Suitable parallel substitution algorithms have been derived by the author; pseudo-code for forward substitution is given in Figure 16. The modifications for backward substitution are simple; the algorithm may be found in [5, Chapter 6]. Subsequent to publication of the author's own research [25], van de Vorst and Bisseling published an algorithm for parallel forward and backward substitution.

The substitution algorithms operate on only one column of the processing array at a time, and the latest version of the relevant vector ( $[z]$  or  $[x]$ ) is passed from column to column as the algorithm proceeds. This is far from the most efficient parallel substitution algorithm possible, since only  $\sqrt{N}$  processors are active concurrently, but has the major advantage of using the same scattered grid distribution as the parallel LU algorithm.

### 3.6 Timing Results

The parallel LU and substitution algorithms described in this section have been implemented by the author in Occam 2 for a transputer array. Detail of the implementation are discussed in [5, Section 5.11]. Preliminary results were presented as [25]. Figure 17 shows efficiencies for a number of different processor array sizes as a function of matrix dimension. The timing results are for single precision runs. The matrix was generated using a simple thin-wire moment method scheme using sinusoidal basis functions and collocation, using results from [26, Section 7.5] for the field radiated by a sinusoidal current. This moment method code was also written in Occam 2. The largest problem solved had 1500 unknowns, and used 25 transputers. The LU solver took about

15 minutes to run, which corresponds to a computation speed of 9.6 MFLOP/s, and an efficiency of close on 90%. The matrix was also generated in parallel and the efficiency of the entire code is very similar to that of the LU part, which is of course the most computationally expensive part. The forward and backward substitution algorithms have also been implemented and despite having rather poor efficiency (as expected), the overall impact on the code is negligible due to the  $O(M^2)$  computational cost of the substitution algorithm.

Figure 18 shows theoretical predictions, which can be seen to be somewhat optimistic, although the general trend is correctly predicted. Reasons similar to those given in Section 2.4 may be advanced for the differences; note that rather finer grain of communication in the LU algorithm is more difficult to model accurately than the communication in the CG algorithm. Recent work by the author indicated that the pipelines have a subtle problem in that the effect of the *set-up* time — the time to initiate a communication on a link — is *not* negligible when elements are being communicated individually; it is around  $6.5\mu s$ , approximately equal to  $t_{comm}$  in single precision. The effect of this is to double  $\beta$  in this case, and this has been incorporated in Figure 18; however, the theoretical results are still some way off the measured results. To permit comparison of the parallel LU and CG algorithms, measurements for a parallel CG algorithm are also shown in Figure 18 for 14 transputers. (The binary tree and mesh topologies cannot use exactly the same number of processors; a tree of 14 and a mesh of 16 is a fair comparison). The CG results were measured with a single precision version of PARNEC. (Note that the results shown previously for the CG solver are for the double precision version of PARNEC.)

Bisseling and Van de Vorst show similar measured results in [20]; the numerical values for efficiency shown in Figure 17 are not directly comparable with their results, which are presumably for real valued matrices, although the latter is not explicitly stated in their paper. The form of the curves is very similar.

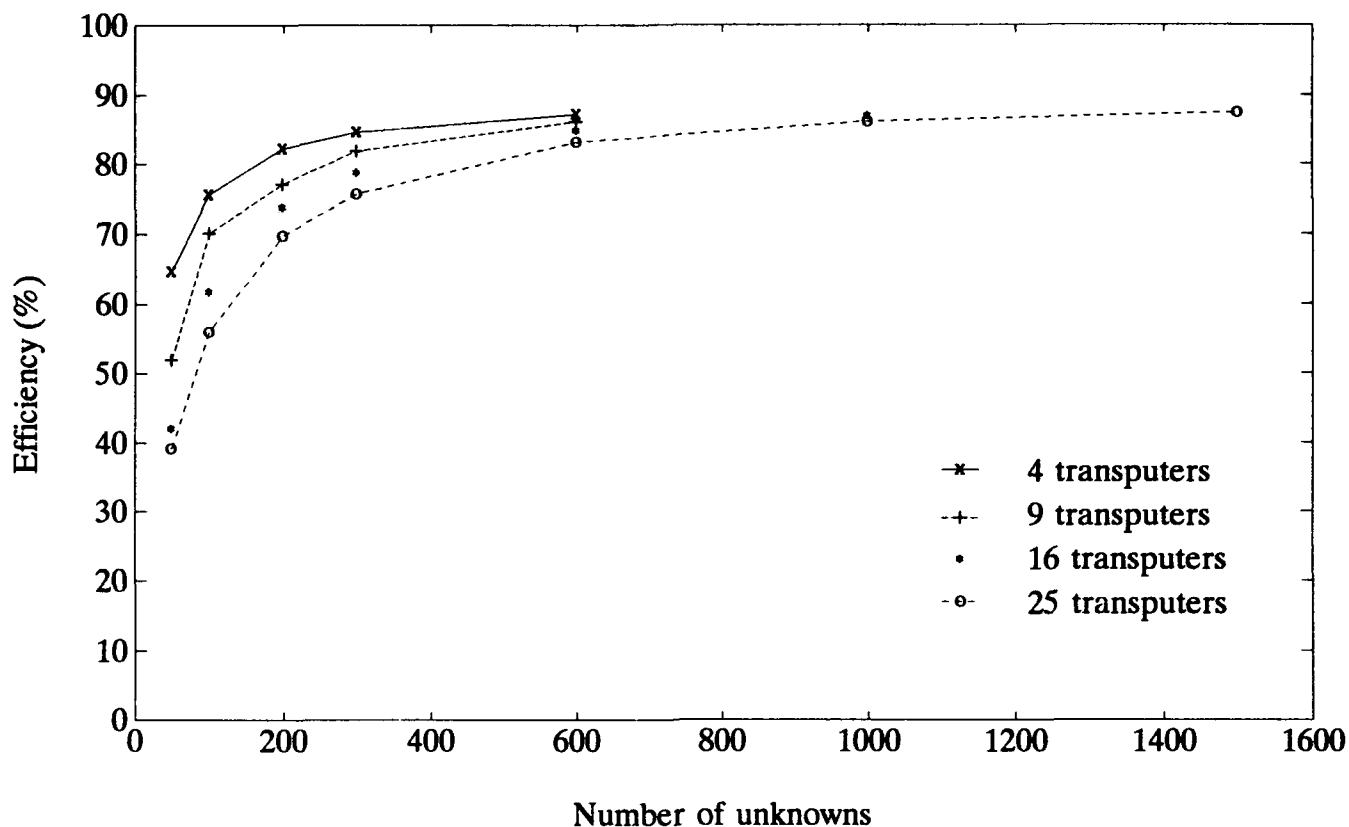
## 4 Scalar efficiencies of the LU and CG algorithms

*Scalar efficiency*<sup>9</sup> deals with the actual run-times of the algorithms when run on the same computer — since the efficiencies of the algorithms discussed in this paper are comparable, it is also very important for these parallel algorithms. It has generally been assumed that using an iterative solver reduces the amount of computation from  $O(M^3)$  for the LU solver to  $n_{iter}O(M^2)$  for the CG solver, where  $n_{iter}$  is the number of iterations required for convergence. The motivation for using parallel iterative solvers for full matrix problems was the expectation that the convergence would be sufficiently rapid for the CG algorithm to terminate in a small number iterations, making the run-time considerably less than the corresponding LU factorization. Iterative methods are widely and successfully used in methods resulting in sparse matrices such as the Finite Element method [27, Chapter 10].

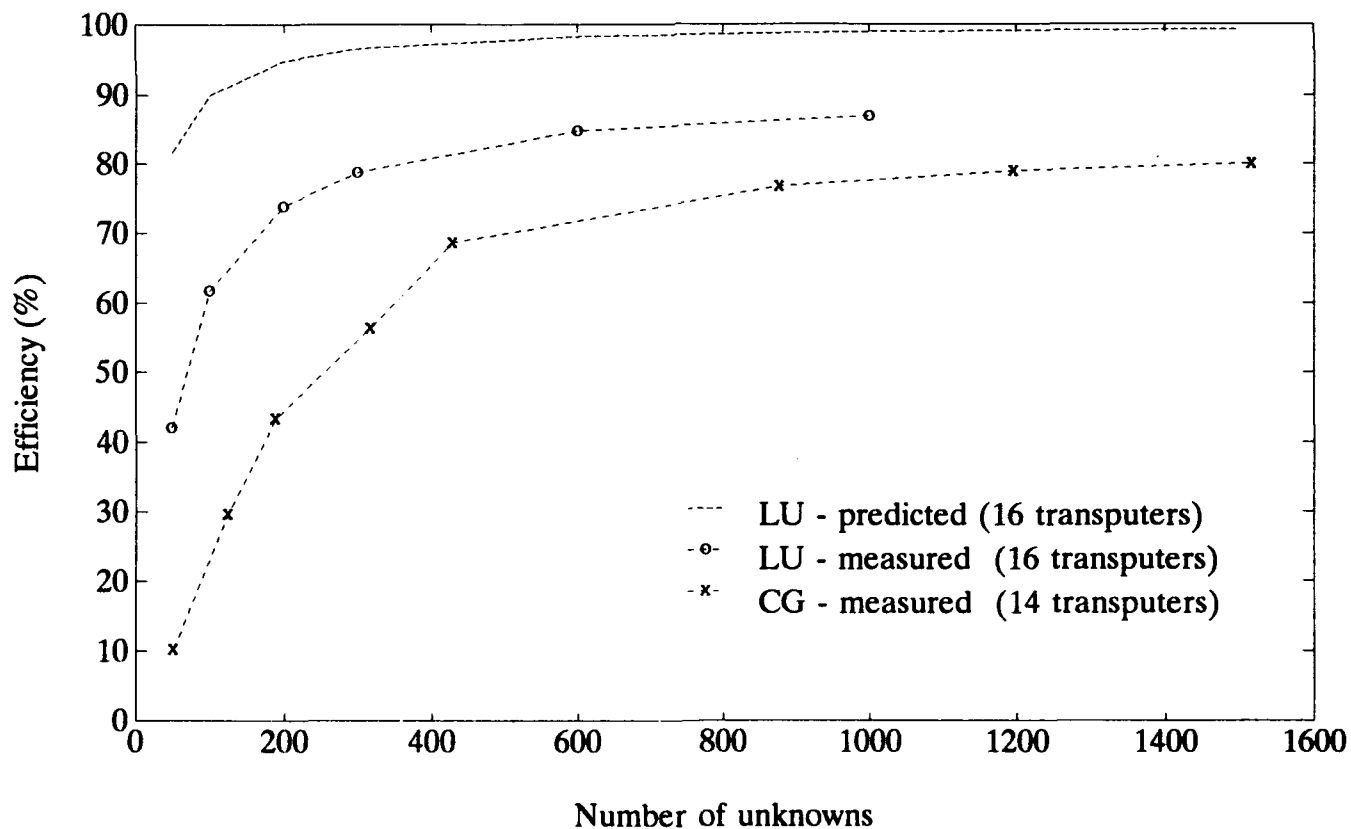
Unfortunately, for arbitrary moment method problems, the number of iterations appears to be a quite substantial fraction of the number of segments, for structures discretized according to some nominal segment length rule, for example  $\lambda/10$ . For problems that are over-discretized, the number of iterations appears to be a function of the problem, and increases only weakly with the discretization, once the structure is satisfactorily discretized. See for example [18]. The reason for this is probably that the extra eigenvalues introduced by the over-discretization are not very significant; see [28]. Unfortunately, it is problems that are just satisfactorily discretized that are frequently of the greatest interest to electromagnetic modellers.

Hence, for the important case of structures just satisfactorily discretized, the computational

<sup>9</sup>The term was suggested by a reviewer, and describes the issue very succinctly.



**Figure 17:** Measured efficiencies of the single precision parallel LU solver.



**Figure 18:** Comparative efficiencies for the single precision LU and CG solvers for similar numbers of transputers.

dependence of the CG algorithm would also appear to be essentially  $O(M^3)$ . Since the efficiencies of both parallel methods are comparable, the serial break-even point can be used, namely where the number of iterations is 1/6 of the matrix dimension. However, even in the largest case investigated to date by the author, using about 2 000 segments, this fraction was closer to 1/4, and was even larger for smaller problems; see Table 7. Peterson and Mittra reported similar results several years ago, for smaller systems with at most a few hundred unknowns [29]. The present author used a normalized error criterion of  $10^{-2}$ , giving an error of around 1%. So, unless one is satisfied with a larger error, the LU method would have been slightly, to considerably, faster for all the problems investigated. The rate of convergence is highly dependent on the problem; for some other work recently performed on relatively large systems (1 000 to 2 000 unknowns), the rate of convergence was much poorer than that mentioned above.

With a multiple right-hand side problem, such as a typical radar cross section problem, the superiority of the LU method has long been acknowledged. The work of Smith *et al.* [30] on using the CG method to solve multiple right-hand sides, by re-using some of the data generated for previous right-hand sides, showed that although significant time savings compared to the standard CG method were possible, for many right-hand sides the LU method remained the better approach. However, a new technique recently proposed by Kastner and Herscovici [31] shows promising results for a multiple right-hand side CG formulation.

## 5 Parallelizing the matrix fill

A number of researchers have reported that the time required for matrix fill, although an  $O(M^2)$  operation, can still dominate moment method codes for large numbers of unknowns [11]. Certainly with a patch code, especially if using the Galerkin formulation, this could be a serious problem. With thin-wire collocation codes such as NEC2, the matrix solve fairly rapidly dominates the matrix fill; an example is shown in Table 7. This is for a CG solver; the number of iterations is also listed. The "break-even" number of iterations where the CG run-time equals that of LU decomposition is  $M/6$ . Using the NEC2 formulation and the CG solver row-block decomposition, the problem of parallel matrix generation was easily solved — one simply decomposes by match point, in precisely the same fashion that the NEC2 out-of-core solver functions. Work is at present in progress on incorporating the LU solver into the parallel NEC2 code.

## 6 Conclusions

### 6.1 General

This paper has presented parallel algorithms for the two algorithms most frequently used in computational electromagnetics for the solution of systems of linear equations. The basic algorithms have been reviewed; parallel algorithms have been presented — both informally and formally in pseudo-code, analyzed, and results obtained with an implementation of the algorithms on a specific parallel computer reported. The experimental results for the CG and LU solvers have been compared both to the theoretical predictions and with each other for similar numbers of processors, demonstrating both theoretically and practically that the parallel LU algorithm presented is more efficient than the parallel CG algorithm shown. The *scalar* efficiency of the LU algorithm is also better since the run-time of the CG method is highly dependent on the rate of convergence of the CG algorithm, and it has been found that the rate of convergence of the CG algorithm for practical problems is not sufficient for the CG algorithm to out-perform the LU algorithm.



Number of segments M	$t_{solve}/t_{fill}$	Number of iterations
50	1.0	14
124	2.2	75
188	2.7	134
316	2.4	131
428	7.2	372
876	9.1	405
1196	10.4	409
1516	11.9	414
1996	21.1	543

**Table 7:** Ratio of the matrix fill to solve times for a particular simulation, viz. a cone-cylinder with four monopoles [5, Section 6.7]. 30 worker transputers were used. All data except for the last entry are for double precision: the 1996 segment data was generated using single precision.

## 6.2 Scaling behaviour and grain size

A very important result has been demonstrated, both theoretically and experimentally, namely the *scaling* properties of the algorithms; larger problems can be solved in an approximately constant time by increasing the number of processors. The scaling property of the parallel LU algorithm considered in this paper has been shown to be better than that of the parallel CG algorithm discussed, although both have quite satisfactory scaling properties. It might be thought that it is self-evident that as the grain size (the number of unknowns per processor,  $n^2$  as used in this paper), increases, so the efficiency will increase — however, this is only a property of an algorithm where the computation cost as a function of the grain size increases faster than the communication cost, and is by no means a general property of all parallel algorithms.

The dependence of the efficiency on the grain has some implications for massively parallel systems that should be considered explicitly. If 50% efficiency is considered acceptable, then a grain size of several hundred is required for acceptable efficiency for the LU algorithm; or put slightly differently, a sub-matrix per processor of dimension twenty or so. An important theoretical result in this paper is that for a given efficiency, this grain size remains constant for the LU algorithm and is only weakly dependent on the number of processors for the CG algorithm (the dependence is approximately  $\sqrt{N}$ ); actual timing results confirm this (Figure 7). Note that since the efficiency is a function of the  $\beta$ , the communication to computation ratio, this break-even point will also be a function of this ratio. For the transputer technology used, this ratio produced very acceptable efficiencies on problems of practical interest: it is, of course, a function of the processor technology, and the user must accept it as a given for a particular processor. For arrays with hundreds of processors, where the algorithms remain relatively coarse-grained, the results in this paper can be extrapolated to show that what are really the classic serial algorithms (albeit in parallel form) can still give very acceptable efficiencies. It should be stated, however, that these results may not apply to truly massively parallel systems, with perhaps tens or hundreds of thousands of processors. The fundamental philosophical issue is that of global interaction (viz. integral equation methods) versus local interaction (viz. differential equation methods) and it is likely that the latter methods with their highly local interaction requirements may be far better candidates for massively parallel computers.

### 6.3 Workstations or transputers?

This section compares T800 based arrays with workstations available at the time of writing (1992) and will inevitably date. As was clearly indicated in the Introduction, the aim of this paper was not to blindly promote transputer arrays; a sober analysis of competing computer technologies is necessary. The present transputer technology (the T800) dates to around the mid-nineteen eighties, and at the time of writing, a contemporary high performance RISC workstation would probably be a better investment; the maximum through-put of a 64 transputer array is about 33 MFLOP/s (in single precision, with a 100% efficiency of the parallel algorithm) as soon as one uses the off-chip RAM, as is typically the case. The author has only been able to use about half this array (25 processors), and obtained 9.6 MFLOP/s. The author has benchmarked the HP720 RISC workstation at close to 20 MFLOP/s on LU decomposition (also in single precision).

Note that this is in not a very fair comparison, since it involves the comparison of computing technologies separated by five to six years; the balance will change back dramatically in favour of transputer arrays when the T9000 is shipped from Inmos [4, 11], so the time invested in developing the parallel algorithms described here is time well invested for future arrays; as serial processors (and the related processors such as super-scalar architectures) increase in speed, so do the individual elements of processor arrays. Cwik and Patterson have reported the accurate solution of what can only be described as *massive* moment method problems with 30 000 unknowns on a 512 node i860 array [32].

### 6.4 Issues still to be addressed

An issue that has not been addressed in this paper is the stability and accuracy of electromagnetically large problems discretized using the moment method. The stability of the LU method, applied to computational electromagnetic problems, has been studied by the author using a thin-wire problem and results obtained indicate that in all except the most exceptional circumstances, involving serious violation of the basic "thin-wire" assumption, the solutions obtained using the LU solver are stable. The availability of a parallel version of NEC2 has permitted the investigation of the accuracy of the moment method for large problems. This was done by using a physically symmetrical problem; first solved with, and then without, exploiting the symmetry. Using symmetry reduces the number of unknowns by the degree of symmetry, thus requiring the solution of a much smaller system of equations. This method has been used to demonstrate the accuracy of NEC2 for problems with up to 2 000 unknowns [5, Chapter 6]. Some preliminary details are to be published in [4].

The CG algorithm was the first major parallel code developed by the author and is not optimal in a number of respects: if pipelining, as exploited in the LU algorithm, were to be exploited in the CG algorithm for the broadcast and gather operations, improvements should be anticipated — this has not been implemented, however. Further, the unparallelized vector operations (addition, subtraction etc.), responsible for the 2.75 factor, could probably also be reduced by parallelizing the vector operations. These amount to "fine-tuning" the existing binary tree algorithm. (One should also bear in mind that given a processor with four communication links, a ternary tree would be more efficient than a binary tree — as mentioned in [3]). However, in the light of the predicted and measured performance of the parallel LU algorithm, an interesting question that arises is whether implementing the parallel CG algorithm on a mesh would result in communication performance similar to that of the parallel LU algorithm. This is a topic for future research. At present, the whole question is possibly more of academic than practical interest, since the existing parallel CG code, while admittedly not optimal, is still very efficient for the problems of interest on presently available arrays. However, rather larger MIMD arrays involving possibly thousands

of processors may require parallel CG algorithms with better scaling properties.

## 6.5 General Conclusions

While the use of more powerful computers with existing algorithms is still ultimately limited by the third power law (see Section 4), for many problems a relatively modest increase in problem size closes the gap between moment method analyses and asymptotic analyses such as the Geometric Theory of Diffraction. The importance of the algorithms discussed in this paper is the good scaling properties that permit the efficient exploitation of large — but possibly not massive — processor arrays for large problems.

## Acknowledgements

The work published here is based on the sections of the author's Ph.D. dissertation, awarded by the University of Stellenbosch. The author would like to acknowledge the contributions of his promoters (Prof. J.H. Cloete, University of Stellenbosch, and Prof. D.A. McNamara, University of Pretoria) to this work. Prof. J.J. du Plessis was instrumental in obtaining the transputer array, and his efforts in this regard are much appreciated by the parallel processing community at this University. The author also would like to thank Prof. Cloete and Messrs. P. Steyn and D.B. Le Roux for their careful proofreading of this paper, and the six anonymous reviewers for raising a number of important points which led to substantial revision of this paper.

## References

- [1] E. K. Miller, "A selective survey of computational electromagnetics," *IEEE Trans. Antennas Propagat.*, vol. 36, pp. 1281–1305, September 1988.
- [2] G. Zorpette (editor), "Special issue: supercomputers," *IEEE Spectrum*, vol. 29, pp. 26–76, September 1992.
- [3] D. B. Davidson, "A parallel processing tutorial," *IEEE Antennas Propagat. Magazine*, vol. 32, pp. 6–19, April 1990.
- [4] D. B. Davidson, "Parallel processing revisited: a second tutorial," *IEEE Antennas Propagat. Magazine*, October 1992. To appear.
- [5] D. B. Davidson, *Parallel Algorithms for Electromagnetic Moment Method Formulations*. PhD thesis, Dept. Electrical & Electronic Engineering, University of Stellenbosch, 1991.
- [6] M. J. Flynn, "Some computer organizations and their effectiveness," *IEEE Transactions on Computers*, vol. C-21, pp. 948–60, September 1972.
- [7] G. C. Fox, M. A. Johnson, G. A. Lyzenga, S. W. Otto, J. K. Salmon, and D. W. Walker, *Solving Problems on Concurrent Processors, Vol I: General Techniques and Regular Problems*. Englewood Cliffs, New Jersey: Prentice-Hall, 1988.
- [8] C. Hafner, "Parallel computation of electromagnetic fields on transputers," *IEEE Antennas Propagat. Society Newsletter*, vol. 31, pp. 6–12, October 1989.

- [9] I. Cramb, D. H. Schaubert, R. Beton, J. Kingdon, and C. Upstill, "Antenna array modelling by parallel processor farms," *Applied Computational Electromagnetics Society Journal*, vol. 6, pp. 143-151, Winter 1991.
- [10] D. C. Nitch and A. P. C. Fourie, "Adapting the Numerical Electromagnetics Code to run in parallel on a network of transputers," *Applied Computational Electromagnetics Society Journal*, vol. 5, pp. 76-86, Winter 1990.
- [11] L. C. Russel and J. W. Rockway, "Application of parallel processing to a surface patch/wire junction EFIE code," *Applied Computational Electromagnetics Society Journal*, vol. 7, pp. 48-66, Summer 1992.
- [12] J. J. H. Wang, *Generalized Moment Methods in Electromagnetics*. New York: John Wiley and Sons, 1991.
- [13] S. L. Ray and A. F. Peterson, "Error and convergence in numerical implementations of the conjugate gradient method," *IEEE Trans. Antennas Propagat.*, vol. 36, pp. 1824-1827, December 1988.
- [14] G. H. Golub and D. P. O'Leary, "Some history of the conjugate gradient and Lanczos algorithms: 1948-1976," *SIAM Review*, vol. 31, pp. 50-102, March 1989.
- [15] A. Jennings, *Matrix Computation for Engineers and Scientists*. Chichester: John Wiley and Sons, 1985.
- [16] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*. Philadelphia: SIAM, 1991.
- [17] C. F. Smith, A. F. Peterson, and R. Mittra, "The biconjugate gradient method for electromagnetic scattering," *IEEE Trans. Antennas Propagat.*, vol. 38, pp. 938-940, June 1990.
- [18] D. B. Davidson and D. A. McNamara, "Comparisons of the application of various conjugate-gradient algorithms to electromagnetic radiation from conducting bodies of revolution," *Microwave and Optical Technology Letters*, vol. 1, pp. 243-246, September 1988.
- [19] R. L. Burden and J. D. Faires, *Numerical Analysis*. Boston: Prindle, Weber and Schmidt, 4th ed., 1989.
- [20] R. H. Bisseling and J. G. G. van de Vorst, "Parallel LU decomposition on a transputer network," in *Parallel Computing 1988: Shell Conference Proceedings, (Lecture Notes in Computer Science series)*, (G. A. van Zee and J. G. G. van de Vorst, eds.), (Amsterdam), pp. 61-77, Springer-Verlag, 1989.
- [21] D. Heller, "A survey of parallel linear algorithms in numerical linear algebra," *SIAM Review*, vol. 20, pp. 740-777, October 1978.
- [22] G. A. Geist, M. T. Heath, and E. Ng, "Parallel algorithms for matrix computations," in *The Characteristics of Parallel Algorithms*, (L. H. Jamieson, D. B. Gannon, and R. J. Douglass, eds.), Cambridge, MA: MIT Press, 1987.
- [23] K. A. Gallivan, R. J. Plemmons, and A. H. Sameh, "Parallel algorithms for dense linear algebra computations," *SIAM Review*, vol. 32, pp. 54-135, March 1990.

- [24] J. G. G. van de Vorst, "The formal development of a parallel program performing LU-decomposition," *Acta Informatica*, vol. 26, pp. 1-17, 1988.
- [25] D. B. Davidson, "Parallel LU decomposition on a transputer array," in *Symposium digest, Vol III, of the 1991 IEEE AP-S International Symposium*, pp. 1500-1503, June 1991. Held in London, Ontario, Canada.
- [26] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. New York: John Wiley and Sons, 1981.
- [27] P. P. Silvester and R. L. Ferrari, *Finite Elements for Electrical Engineers*. Cambridge: Cambridge University Press, 2nd ed., 1990.
- [28] A. F. Peterson, C. F. Smith, and R. Mittra, "Eigenvalues of the moment-method matrix and their effect on the convergence of the conjugate gradient algorithm," *IEEE Trans. Antennas Propagat.*, vol. 36, pp. 1177-1179, August 1988.
- [29] A. F. Peterson and R. Mittra, "Convergence of the conjugate gradient method when applied to matrix equations representing electromagnetic scattering problems," *IEEE Trans. Antennas Propagat.*, vol. 34, pp. 1444-1454, December 1986.
- [30] C. F. Smith, A. F. Peterson, and R. Mittra, "A conjugate gradient algorithm for the treatment of multiple incident electromagnetic fields," *IEEE Trans. Antennas Propagat.*, vol. 37, pp. 1490-1493, November 1989.
- [31] R. Kastner and N. Herscovici, "A concise conjugate gradient computation of plate problems with many excitations," *IEEE Trans. Antennas Propagat.*, vol. 38, pp. 1239-1243, August 1990.
- [32] T. Cwik and J. Patterson., "The solution and numerical accuracy of large MoM problems," in *Symposium digest of the 1992 URSI Radio Science Meeting*, p. 335, July 1992. Held in Chicago, Illinois, USA.

---

# **'A Priori' Knowledge, Non-Orthogonal Basis Functions, and Ill-Conditioned Matrices in Numerical Methods**

Ch.Hafner, Swiss Federal Institute of Technology, Zurich

## **Abstract**

Many terms and ideas used in numerical methods have their origin in analytical mathematics. Despite the well-known discrepancies between number spaces of computers and those of good old mathematics, the consequences of applying mathematical theorems to numerical methods and the importance of physical reasoning are often underestimated. The objective of this paper is to demonstrate that introducing 'a priori' knowledge of a problem into a numerical code can lead to superior numerical techniques but it may violate analytic dogmas at the same time.

## **Introduction**

It was essential for Isaac Newton to benefit from introducing the infinitesimal calculus into physics. Although Newton himself developed concepts of matter close to post-modern fractals, the infinitesimal calculus forced people to consider space, time, and functions of space and time to be continuous. "Real numbers" that were believed to represent chaos (Johannes Kepler) replaced integers and Euclids geometry. Maxwell theory was the culmination of the idea of continua: Space, time, field, everything was considered to be continuous and extending to infinity. Thus, Maxwell's theory was much more consistent than Newton's theory in which the mass points did not really fit the idea of continuity. Although this convinced Einstein to develop even more powerful field theories, problems soon arose. Integers struck back. This was the beginning of quantum mechanics with its strange finite number spaces and probability concept. Ironically, the computer is unable to directly understand Maxwell theory, the most important theory of electromagnetics that still is essential for the construction of today's computers. Real numbers, infinity, continuity, and random numbers are far away from computer architecture. Thus, one cannot exactly apply most of the well-known theorems in classical mathematics to develop algorithms for computers. Such theorems can even be misleading.

The first mystery in computational mathematics is the following. Analytically one usually works in function spaces with an infinite number of dimensions. The number of dimensions is often uncountable. For computers, this number must be finite and preferably relatively small. I.e., one has to omit an infinite number of dimensions. Nonetheless, one can obtain useful results. For example, the number of basis functions or the dimension of the function space of a Fourier integral is uncountable infinite, whereas the number of basis functions of Fourier series is countable infinite. Obviously, Fourier integrals cannot be evaluated numerically, except when the function behaves well, i.e., is sufficiently simple. At first glance, it seems that nature is so kind with us, that we often can use Fourier

---

integrals. But in fact, we usually simplify our models of nature to such an extent that our methods work. For example, we know that absolutely flat surfaces do not exist in real world, but we work with such surfaces in most of the numerical models. Again, nature is really kind: We obtain useful results.

If we consider the approximation of continuous functions, flat surfaces, etc. by computers, we find another mystery. The approximations of real numbers, continuous functions, surfaces, etc. made by computers are rough rather than continuous or flat. The situation can be illustrated by Newton's mirror. To obtain a mirror, Newton polished metal. He knew that the surface initially is rough. Thus, the rays of light are reflected in very different directions. He did recognize that polishing made the surface less rough but never eliminated the roughness completely. Thus, the rays of light should always be reflected in very different directions and one should never obtain a mirror. Newton did introduce the concept of a kind of fluid on the surface of the metal. The rays are reflected by this fluid rather than by the surface of the metal. This explained why there are mirrors. Since Maxwell, we "know" that this is the effect of the wavelength. However, the computer approximates flat surfaces in the analytic model by rough surfaces. Although this is an essential discrepancy, we get useful results.

Of course, we cannot do whatever we want for obtaining useful results. Provided that we are not pure mathematicians, the results we are looking for have a certain meaning and usually are compared with measurements. If we consider the huge difference between measurement, analytical model, and numerical computation, the fact that we can get useful results again is a miracle. However, it is very important to note that we often introduce some 'a priori' knowledge in order to obtain useful results. This can be done to discard wrong results like spurious modes but also in the modelling and in the design of codes. For example, all adaptive methods use a certain kind of 'a priori' knowledge, i.e., the data obtained from previous computations. Since it is such a big miracle that we are able to simulate measurements by computations, the 'a priori' knowledge is extremely important. It can even be used for entirely removing the theory, e.g., Maxwell theory, from the code. This is obviously true for heuristic codes, but one can also try to implement codes that directly analyze measurements, find a theory, simulate, and predict measured data. However, the power of 'a priori' knowledge should not be underestimated. Numerical codes that ignore 'a priori' knowledge (some mathematicians might like such codes) turn out to be inefficient in most cases. As people should be able to learn from each other or from previous generations, a numerical code should at least take advantage of some a priori knowledge of its designer or of its user.

In many series expansions, for example, Fourier series, the basis functions are ordered. This is important for counting the basis functions and above all for defining the convergence. It is well known that convergence is important for the efficiency of a numerical method, but the analytic proof of convergence for any series expansion is not sufficient in practice. Instead, we need a sufficiently fast convergence. On the other hand, one can also obtain useful numerical techniques with asymptotic and other series that do not converge at all. When 'a priori' knowledge is considered, one often can eliminate some of the basis functions in ordered series. For example, symmetries often lead to such reduced sets of basis functions. But one can even obtain completely different, somewhat "chaotic" series expansions. For such expansions, the term "convergence" sometimes cannot even be defined. Nonetheless, they can be very powerful.

The basis functions of many series expansions are orthogonal. The orthogonality considerably simplifies the computation of the coefficients of such series. Thus, destroying the orthogonality seems to be a sacrilege. It is important to note that orthogonality always depends on the definition of a scalar product. In computational electromagnetics, the definition of the scalar product depends on the modelling and can be different from the scalar product with respect to that the basis functions are orthogonal. Although the basis functions are orthogonal in a certain sense, they are often non-orthogonal with respect to the scalar product that actually is used\*. Orthogonalization procedures are very time consuming. Therefore it is not reasonable to use them. Numerically, working with non-orthogonal functions is quite simple, provided that the functions are not almost linearly dependent. One usually obtains matrix equations that can be solved with many well-known algorithms. Again, analytic mathematics created the term of linear dependence. However, the requirement of linearly independent basis functions in a numerical code is not enough in general. It is quite cumbersome to detect "almost dependent" basis functions. A well-known measure is the condition number of the matrix. A large condition number means that the matrix is ill conditioned, whereas the "optimal" condition number (which is one) can be obtained when orthogonal basis functions are involved. It will be demonstrated in the following sections that one can obtain more accurate results by increasing the condition number. There are two "optimal" condition numbers: one that only considers the matrix and another that considers the results. The latter can be considerably bigger than the former. Thus, improving the condition of a matrix or using even orthogonal functions may have undesired effects.

## Intelligent Fourier Analysis

Fourier integrals usually are applied to time-dependent functions defined in the interval  $-\infty < t < \infty$ . It is considered to be true that the "real" time of our universe has finite upper and lower limits. Every measurement has even much more restrictive upper and lower limits. In such a finite time interval, Fourier integrals can be replaced by the much more simple Fourier series. In fact, the harmonic functions used as basis functions in the Fourier integrals are orthogonal provided that an appropriate scalar product is defined on the interval  $-\infty < t < \infty$ . On a finite interval, no scalar product can be defined in such a way that all basis functions used in the Fourier integral are orthogonal, whereas it is no problem to find a scalar product that makes the basis functions of the Fourier series

---

\* For example, the functions  $r^n \cos n\varphi$  and  $r^n \sin n\varphi$ , where  $r\varphi$  are polar coordinates and  $n$  is an integer number, are orthogonal if the scalar product is defined as  $(f, g) = \int_{r=0}^{\infty} \int_{\varphi=0}^{2\pi} f g r dr d\varphi$ . The functions  $r^n \cos n\varphi$  and  $r^n \sin n\varphi$  can be used for solving the Dirichlet problem in a bounded 2D domain  $D$ . Depending on the numerical method, a scalar product is applied that is defined either on the domain  $D$  or on its boundary  $\partial D$ . For example,  $(f, g) = \int_{\partial D} f g ds$ . With respect to such a scalar product, the functions are non-orthogonal, except when  $\partial D$  has a very special shape.



orthogonal\*. Obviously, the basis of Fourier series is obtained from the basis of Fourier integrals by erasing most of the basis functions, i.e., the biggest part of the spectrum in such a way that a discrete spectrum is obtained. Thus, one has the same effect as when one takes very high symmetries (symmetry groups with an infinite dimension) into account for any series expansion.

It is obvious that the numerical computation of Fourier series is much simpler than the one of Fourier integrals. There seems to be no reason for Fourier integrals since neither "real" time nor the time intervals of measurements are infinite. But if one considers the spectra of practical functions, one often finds a behavior that can much better be approximated by Fourier integrals (with a practically limited spectral domain). Thus, the approximation of functions in a finite time interval by functions defined on an infinite time interval often is reasonable.

Let us now assume that a function  $f$  is defined in the interval  $-\infty < t < \infty$  and  $f$  can be expanded by a Fourier integral with a finite spectrum. If this function is measured in a time interval  $0 < t < T$ , it can be expanded by a Fourier series. If we assume that the measured function  $f^0$  is known exactly in every point of the interval, the spectrum of the Fourier series turns out to be discrete and infinite, i.e., the spectra of  $f$  and  $f^0$  turn out to be completely different. If we compute the Fourier series of  $f^0$  outside the interval  $0 < t < T$ , we find that this is a periodic function with the period  $T$ . Thus, the "true" function  $f$  and the expansion of the measured function  $f^0$  are completely different outside the interval  $0 < t < T$  even in the best case, where the measurement of  $f^0$  is exact in the entire interval. As a consequence, the Fourier series cannot be used to predict the behavior of  $f$  for  $t > T$  even when  $t$  is not much bigger than  $T$ , as the following discussion illustrates.

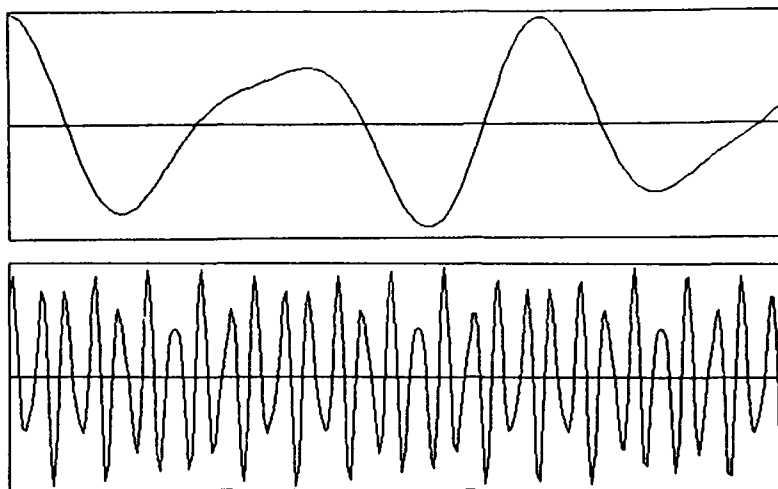
For reasons of simplicity, we now assume that  $f$  simply consists of two harmonic functions, for example,

$$f = A \cos \omega_A t + B \cos \omega_B t.$$

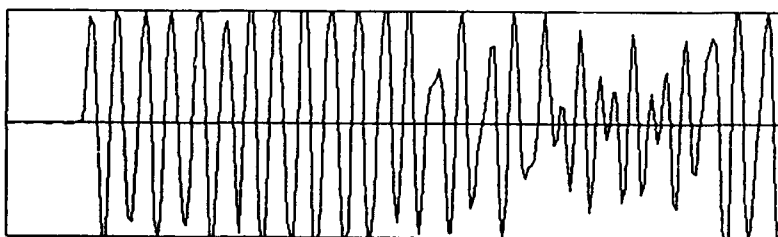
(See figure 1.) Note that  $f$  is not periodic at all when  $\omega_A/\omega_B$  is irrational. If we measure  $f$  in a finite number of points  $k\Delta t$ ,  $k = 1, 2, \dots, K$ , we easily can approximate it by a Fourier series. Usually, we will not be so lucky that both frequencies  $\omega_A$  and  $\omega_B$  are in the discrete spectrum of the Fourier series. Nonetheless, from a pure mathematical point of view, everything is fine: 1) The basis functions of the Fourier series are orthogonal with respect to an appropriate scalar product defined on the interval  $0 < t < T$ . 2) The condition number of the corresponding matrix is one. 3) The system of equations can be solved with any algorithm. 4) Iterative matrix solvers converge in only one step, and so on. But there are two important drawbacks: 1) The convergence is quite bad in most cases. 2) The prediction of  $f$  outside the interval of the measurement is completely useless. Figure 2 illustrates this.

---

\* This is quite clear because the number of independent basis functions is infinite but countable – otherwise, the set of basis functions of the Fourier series would be incomplete. Since the number of basis functions of a Fourier integral is uncountably infinite, there must be uncountably many dependent functions, and dependent functions never are orthogonal.



**Figure 1** Function  $f(t) = \cos(18\pi t) + 0.4 \cos(28\pi t)$ . Top: in the interval  $t = 0 \dots 2$  used for sampling and expanding it. Bottom: In the time interval  $t = 0 \dots 20$  used for computing the error function.



**Figure 2** Error obtained when a Fourier series is applied to approximate  $f(t) = \cos(18\pi t) + 0.4 \cos(28\pi t)$ . Note that the condition number of the matrix is one.

If we want to apply Fourier integrals with a finite spectrum to approximate the same function measured in the same points, we recognize that we do not know how to choose the limits of the spectrum. Moreover, the spectral domain needs to be discretized for numerical integration. Finally, the basis functions are no longer orthogonal on the interval  $0 < t < T$ . Here, some 'a priori' knowledge, for example, an estimation of the limits of the spectrum, can be very helpful.

When we do not have any 'a priori' knowledge or when we are too lazy to care about it, the computer can try to get it from analyzing  $f^0$ . An idea how this can be achieved is the following: 1) Assume that  $f^0$  can be approximated by a single harmonic function and compute its amplitude, frequency  $\omega_1$ , and phase. The computation of the frequency requires a nonlinear optimization. Moreover, a least squares procedure (best fit) is reasonable here. 2) Since the frequency estimated in the previous step probably is inaccurate (this is often typical for 'a priori' knowledge), use now several frequencies  $\omega_{1i}$  which are close to  $\omega_1$ . The differences  $\omega_{1i} - \omega_1$  depend on the accuracy of the estimation of  $\omega_1$ . The estimation of this accuracy is another problem that is not considered here. 3) Approximate

$f^0$  by the series

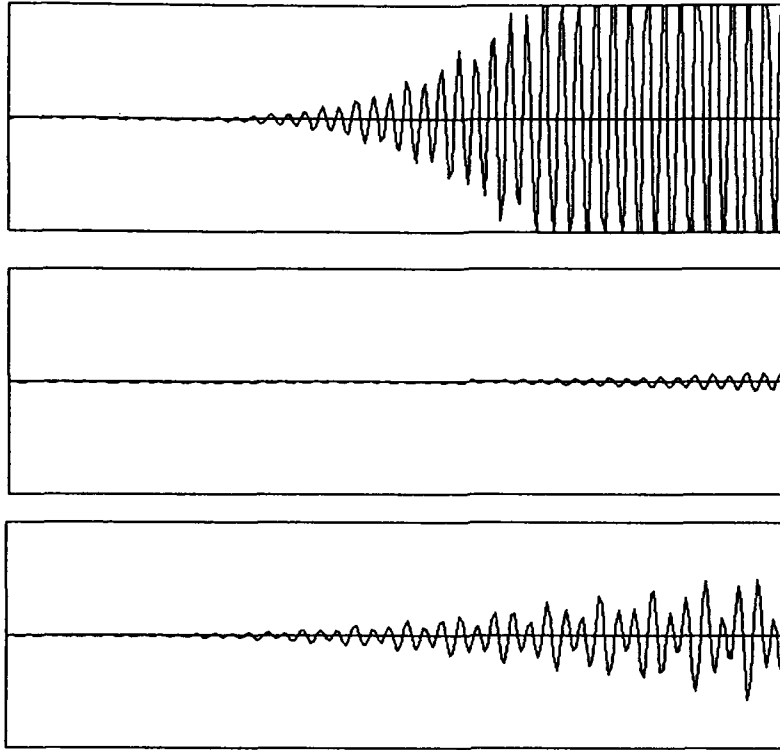
$$f^1 = \sum_i^I A_{1i} \cos \omega_{1i} t + B_{1i} \sin \omega_{1i} t.$$

Note that the basis functions in this expansion are almost linearly dependent and that the condition number of the matrix to be solved turns out to be very bad when the differences  $\omega_{1i} - \omega_1$  are very small. Obviously, the better  $\omega_1$  is estimated, the worse the situation becomes, i.e., the more accurate the 'a priori' knowledge, the worse the condition of the matrix. Thus, it is extremely important that the method for computing the unknowns does not fail when the condition is bad. Here, it is strongly recommended to use Generalized (weighted) point matching [1,2] with an overdetermined system of equations that is directly solved with QR decomposition or even singular value decomposition. 4) Compute the error function  $e = f^0 - f^1$  and analyze  $e$  as you did analyze  $f^0$ , i.e., estimate the frequency  $\omega_2$ , add a set of basis functions  $\cos \omega_{2i} t$ ,  $\sin \omega_{2i} t$ , compute the parameters in the series expansion. Of course, this procedure can be repeated. Since we only have two harmonics in our original function  $f$ , this is not necessary here. The figures 3 and 4 illustrate this for 3, 5, and 7 basis functions per frequency. Note that one can not only analyze the function and errors in the time domain but also the spectrum in the frequency domain. From the latter one often can obtain more accurate estimations of the frequencies which directly leads to an iterative improvement of the results. The biggest advantage of this procedure is the fact that one can predict  $f$  outside the interval where it has been measured. Moreover, the number of basis functions required can considerably be reduced and the accuracy of the approximation in the interval  $0 < t < T$  is much better than for the Fourier series. Concerning the condition of the matrices, the following behavior is important: when the number of basis functions  $\omega_{ki}$  per estimated frequency  $\omega_k$  is increased for a fixed maximal difference  $|\omega_{ki} - \omega_k|$ , the condition number is increased but the results obtained are at first improved. The point where the results become worse depends on the problem but also on the numerical method used for computing the parameters in the series expansion. At this point the condition can be very bad, when a good numerical technique is applied. Maybe, the badness of the condition is even a measure for the quality of the method.

Note that the function  $f(t) = \cos(18\pi t) + 0.4 \cos(28\pi t)$  used for testing the "intelligent" procedure is periodic because  $18/28$  is rational. Thus, one can apply a Fourier series. This is very successful when one knows the actual length of the period  $T_a$ . Since  $T_a$  is much larger than the interval  $T$  where the function is measured, the basis functions of the Fourier series are non-orthogonal with respect to a scalar product defined on the interval  $t = 0 \dots T$  and the matrix turns out to be ill-conditioned. Of course, the knowledge of  $T_a$  is an 'a priori' knowledge as well.

## Computational Electromagnetics with the MMP Code

Computational electromagnetics is a considerably more complex task than the approximation of functions by a series expansion with a given set of basis functions. But essentially most of the codes for computational electromagnetics use either explicitly or implicitly a



**Figure 3** Error obtained when the "intelligent" procedure is applied to approximate  $f(t) = \cos(18\pi t) + 0.4 \cos(28\pi t)$ . Top: Two times three non-orthogonal basis functions. The condition number of the matrix is 425. Middle: Two times five non-orthogonal basis functions. The condition number of the matrix is  $9.7E6$ . Bottom: Two times seven non-orthogonal basis functions. The condition number of the matrix is  $1.0E8$ . All computations with single precision.

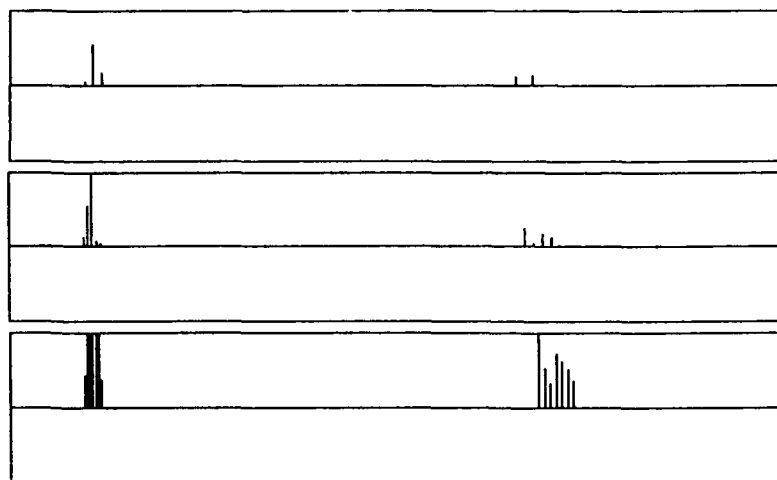
similar expansion of the electromagnetic field. Thus, one can find similar effects as shown in the previous section.

The MMP code [3,4] is very closely related to analytic solutions of the Maxwell equations. In each domain, the field  $F$  is approximated by

$$F = F^0 + \eta = \sum A_i F_i + \eta$$

where  $\eta$  is an unknown error function,  $A_i$  are the unknown parameters to be computed, and  $F_i$  are known solutions of Maxwell equations in the corresponding domain, playing the role of basis functions. Obviously, the approximated field  $F^0$  automatically fulfills Maxwell's equations in the corresponding domain and the parameters  $A_i$  have to be computed in such a way that the boundary or continuity conditions are fulfilled numerically. In the MMP code, multipole fields are preferred basis functions  $F_i$  but many other functions are available as well. It is well known that the quality of the results depends not only on the basis functions but also on the technique used for computing the parameters.

It is very clear that simple techniques can be applied when the basis functions are orthogonal. Actually, multipole functions are orthogonal when a scalar product is defined



**Figure 4** Spectrum obtained when the “intelligent” procedure is applied to approximate  $f(t) = \cos(18\pi t) + 0.4 \cos(28\pi t)$ . Top: Two times three non-orthogonal basis functions. Middle: Two times five non-orthogonal basis functions. Bottom: Two times seven non-orthogonal basis functions. Note that the large amplitudes in the last spectrum indicate that there are severe cancellations in this computation.

everywhere in space. But this not relevant for the numerical computation of the parameters since the boundary conditions do not hold everywhere in space. When a scalar product is defined on the boundaries of the domains, the multipole functions are no longer orthogonal – except for very special and simple cases. Thus, one has to deal with non-orthogonal basis functions anyway. Unfortunately, Simple MultiPole (SMP) expansions that have successfully been used for relatively simple geometries [5] do not converge rapidly in most cases and are useless for more complex geometries. To overcome this drawback, Multiple MultiPole (MMP) expansions have been proposed [6] whereby most of the scientists discarded the SMP approach in favor of the Method of Moments (MoM) [7].

In older codes based on SMP expansions the parameters have been computed using Simple Point Matching SPM on the boundaries. It is very important to recognize that SPM is a relatively weak method that does not work unless the matching points are selected appropriately. There seems to be a relatively strict relation between the basis functions and the appropriate locations of the matching points. If SMP expansions are replaced by MMP expansions, it is extremely hard to find appropriate locations for both the multipoles and the matching points. This is a reason for abandoning not only SMP but also SPM and replacing it by defining a scalar product and performing a projection on a certain set of testing functions. People working with MoM have noticed that there is some relation between the numerical computation of the scalar products and the matching

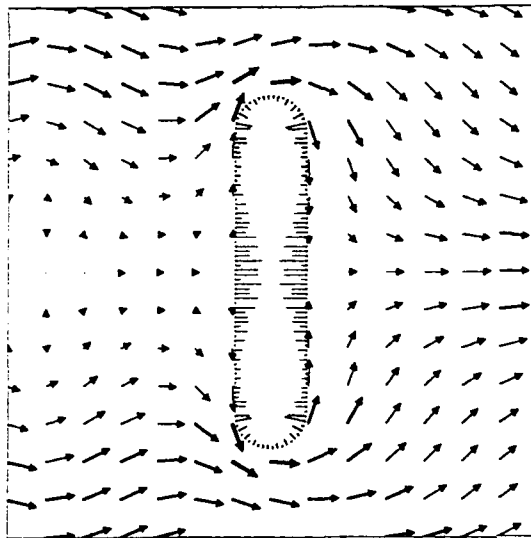
points [2]. When one uses more points to compute the integrals in the scalar products than the number of unknown parameters, one obtains an equivalence with the weighted point matching technique. This technique is essentially the same as the Generalized Point Matching (GPM) that has already been used in the early versions of the MMP code [8] for removing the problems of SPM with MMP expansions. In GPM the numerical equivalence mentioned above has been used for deriving an appropriate geometrical weighting of the equations. In addition, a physical and a user defined weighting is implemented in the MMP code, and continuity equations for all components of the electric and magnetic field are usually imposed.

Although GPM is numerically equivalent to a projection method with trapezoidal numerical integrations and Galerkin's choice of testing functions [1], it is very important to note that the overdetermined system of equations  $AP = G$  obtained from GPM is solved directly in the MMP code using a Givens updating based on QR decomposition, whereas the projection method usually applied in MoM codes leads to the square system  $A^*AP = A^*G$ . Solving  $AP = G$  directly is numerically superior to solving  $A^*AP = A^*G$ . In fact, the latter is useless when the condition of the matrix  $A$  is not sufficient. Mathematicians might believe that this is no drawback because ill-conditioned matrices should be avoided anyway. But we did already see in the previous section that one can obtain numerically more accurate results with "worse" matrices with larger condition numbers. Exactly the same effect can be shown using the MMP code.

Actually, the computation of the condition number of a matrix is quite cumbersome. Some algorithms like Cholesky decomposition [9] allow to estimate it but this estimation is very inaccurate and it is always too high. Singular value decomposition can be performed to compute the condition number accurately. This requires the storage of relatively large matrices and is quite time consuming. For these reasons, singular value decomposition is not implemented in the usual version of the MMP code, but it is contained in a testing version of the 2D MMP code. In addition to the singular value decomposition, the columns of the matrix  $A$  must be scaled. The column scaling does not affect the results in most cases – in some cases it even leads to slightly worse results because of the additional numerical operations – but it is important for iterative matrix solvers and for the condition number. For example, when the scattering of a plane wave from a circular cylinder is computed with the MMP code, the basis functions are orthogonal and the condition number is one if the columns of the matrix are scaled. Otherwise, a typical condition number is 10000 and depends on the frequency and size of the cylinder.

For demonstrating that the accuracy can be improved when ill-conditioned matrices are used, a 2D counterpart of the famous 3D ACES cylinder [10] has been considered with a vertically incident plane wave. A cylinder with a height of one wavelength and a width of 0.4 wavelengths (see figure 5) has been computed with 2D MMP. The matrix to be solved had 82 rows and 45 columns. I.e., the system was slightly overdetermined. On the long symmetry axis of the cylinder,  $M$  multipoles were set with equal distances between their origins, according to the MMP rules discussed in [1]. When an SMP expansion, i.e., only one multipole, was used, the result turned out to be completely wrong because of the "wrong" locations of the matching points. But for more multipoles, useful results have been obtained (see table). Note that the orders used for the different multipoles have been varied in such a way that always 45 unknowns have been obtained. Of course, both the condition number and the error depend on the distribution of the orders of the poles but this problem is not considered here. In the computation of the example, the horizontal

symmetry axis has been used for reducing the computation time. One multipole has always been set in the center. Because of the symmetry operations one effectively has  $2M - 1$  multipoles when one sets  $M$  multipoles with one multipole on the axis. This explains why only odd numbers are contained in the table below. The error number is computed by the MMP code and contains the mismatching of all field components on the boundary. It is a relatively reliable measure and usually turns out to be considerably higher than the error estimated by other codes, for example, [11].



**Figure 5** Scattering at a 2D “ACES-shape” cylinder used as an example to test the condition numbers and the MMP errors for different MMP expansions. Time average of the Poynting vector field and error distribution in the matching points for a total number of 11 multipoles. The condition number  $5.5E5$  is already quite high but obviously the results are very accurate. Computation with double precision.

Condition Number and Error in Function of Multipoles for a 2D Conducting ACES-Shape Cylinder		
Multipoles	Condition Number	MMP Error
3	2.6E2	5.4E-1
5	7.8E2	9.3E-3
7	1.3E4	5.2E-3
9	1.3E5	2.2E-3
11	5.5E5	2.9E-3
13	2.4E6	3.7E-3

---

If one considers the table above, one will recognize that the optimal result is obtained with 9 multipoles where the condition is already quite bad. Since the condition number is considerably increased between 5 and 7 multipoles, a code that is not able to handle ill-conditioned matrices might lead to optimal results when only 5 multipoles are used. Since the example is very simple, the condition numbers remain moderate. For more complex applications, very high condition numbers can be obtained. In this case, even QR and singular value decomposition fail. But good results have been obtained with block-iterative matrix solvers [12].

Note that 'a priori' knowledge is very important in the MMP code. 1) Some 'a priori' knowledge is used in the modelling, where the user defines the matching points. The GPM in the MMP code allows to set much higher matching point densities near critical points, i.e., points where the user assumes 'a priori' that the field varies considerably. This would not be possible or at least not to same extent with SPM. 2) The setting of the multipoles can be done with 'a priori' knowledge. Many simple rules have been established for this purpose and the graphic MMP editors contain semi-automatic procedures for the pole setting. 3) When a rough model has already been computed, one can use a lot of 'a priori' knowledge for improving the model. In this case, the MMP code even allow to introduce the field computed from a previous run (eventually with a different model) as a new basis function called "connection" [13,1]. Connections can lead to excellent results with ill-conditioned matrices.

## Conclusion

It has been demonstrated that terms known from analytic considerations and goals like orthogonal basis functions and small condition numbers of matrices can be misleading and prevent engineers from designing useful codes for computational electromagnetics and similar tasks. Introducing 'a priori' knowledge in numerical codes requires open structures and often leads to ill-conditioned matrices. Thus, it is important to develop and apply methods for handling such matrices, for example, the generalized point matching used in the MMP code instead of the projection technique used in many MoM codes.

Mathematicians usually derive theorems and algorithms for certain classes of functions. For them, all possible solutions are of the same interest. Engineers often look for very special, physically meaningful solutions. Although algorithms able to approximate any solution can naturally be used for approximating special solutions, they are inefficient compared with more specialized or more intelligent algorithms. Similarly, mathematical theorems can be useless for engineers. An engineer who wants to simulate, for example, the scattered field for a certain geometry and for a given incident wave, has to find a relatively small set of basis functions allowing to approximate the solution with the desired accuracy. A mathematical proof of the completeness of a certain set of infinitely many basis functions allowing to approximate, for example, all regular solutions for any incident wave and for any geometry is neither necessary nor helpful. The important question "how many basis functions are required to solve a given problem with the desired accuracy?" is never answered by mathematicians. This big discrepancy between classical mathematics and



---

engineering forces us to invent new, efficient, and intelligent codes that are not based on a "solid" mathematical foundation.

Design of intelligent codes does not only mean to gather a lot of knowledge from books and papers but also to violate analytic dogmas for testing whether they still hold in the age of computers.

## References

- [1] Hafner, Ch., *The Generalized Multipole Technique for Computational Electromagnetics*, Artech House, Boston, 1990.
- [2] Djordjevic, A.R. and Sarkar, T.K., "A Theorem on the Moment Methods", *IEEE Trans. on Antennas and Propagation*, Vol. AP-35, March 1987, pp. 353-355.
- [3] Hafner, Ch., *2D-MMP; Two Dimensional Multiple Multipole Software and User's Manual*, Artech House, Boston, 1990.
- [4] Hafner, Ch. and Bomholt, L., *The 3D Electrodynamic Wave Simulator*, to be published by John Wiley and Sons, 1993.
- [5] Goell, J.E., "A Circular-Harmonic Computer Analysis of Rectangular Dielectric Waveguides," *Bell System Technical Journal*, Sept. 1969, pp. 2133-2160.
- [6] Hafner, Ch., *Beiträge zur Berechnung elektromagnetischer Wellen in zylindrischen Strukturen mit Hilfe des 'Point-Matching'-Verfahrens*, Diss. ETH, No. 6683, Zürich, 1980.
- [7] Harrington, R.F., *Field Computation by Moment Methods*, New York: Macmillan, 1968.
- [8] Ballisti, R. and Hafner, Ch., *The Multiple Multipole Method (MMP) in Electro- and Magnetostatic Problems.*, *IEEE Trans. on Magnetics*, Vol. 19, 1983.
- [9] Dongarra, J.J., C.B. Moler, J.R. Bunch, and G.W. Stewart, *Linpac Users Guide*, Philadelphia: SIAM, 1979.
- [10] Ludwig, A.C., Kuster, N., Glisson, A., and Thal, A., "5:1 Dipole Benchmark Case", *The ACES Collection of Canonical Problems, Set 1*, Applied Computational Electromagnetics Society, 1990, pp. 34-59.
- [11] Ludwig, A.C., "A New Technique for Numerical Electromagnetics," *IEEE AP-S Newsletter*, Vol. 31, February 1989, pp. 40-41.
- [12] Kuster, N., *Dosimetric Assessment of EM Sources Near Biological Bodies by Computer Simulations*, Diss. ETH, No. 9697, Zürich, 1992.
- [13] Bomholt, L., *MMP-3D — A Computer Code for Electromagnetic Scattering Based on the GMT*, Diss. ETH, No. 9225, Zürich, 1990.

ON THE CONVERGENCE OF THE METHOD OF MOMENTS, THE BOUNDARY-RESIDUAL  
METHOD, AND THE POINT-MATCHING METHOD WITH A RIGOROUSLY  
CONVERGENT FORMULATION OF THE POINT-MATCHING METHOD

K. J. Bunch and R. W. Grow  
Microwave Device and Physical Electronics Laboratory  
Department of Electrical Engineering  
University of Utah  
Salt Lake City, Utah 84112

ABSTRACT

The method of moments reduces to the boundary-residual method or the point-matching method with a suitable weighting function. This paper shows another means by which these three methods can produce equivalent results. Arguments are given as to why point matching can fail to converge, while the other two methods rigorously converge. An example is given to support these arguments.

EQUIVALENCE OF METHODS

The method of moments [1], the boundary-residual method [2, 3], and the point-matching method [4] are three seemingly different methods for field computation. Harrington [1] has shown, however, how the method of moments encompasses the other two methods through the proper selection of weighting functions. Another means exists by which all three methods can become computationally equivalent.

Consider the problem posed from the perspective of the method of moments [1]. A deterministic equation such as

$$L \sum_i \alpha_i f_i(s) = g(s) \quad (1)$$

is to be solved over some range  $s$ . The equation, as it applies to electromagnetics, may satisfy the boundary conditions of a particular problem, e.g., the continuity of the tangential fields across the boundary. The summation then represents the field within a region, and the operator  $L$  produces the tangential fields at the boundary  $s$ .  $g(s)$  is the value of the tangential fields from, say, the known incident field. A weighting function  $W_i$  can be multiplied on both sides of Eq. 1 and integrated over the boundary  $s$  to produce a matrix equation:

$$\vec{M} \vec{\alpha} = \vec{g} \quad (2)$$

with

$$\bar{M}_{ij} = \int_s W_i(s) L f_j(s) ds \quad (3)$$

$$g_i = \int_s W_i(s) g(s) ds \quad (4)$$

The boundary-residual method can be derived from these equations by setting the weighting functions equal to

$$W_i(s) = (L f_i(s))^* \quad (5)$$

where \* denotes the conjugate operator. The truth of this assertion can be shown by defining the residual along the boundary,

$$R(s) = L \sum_i \alpha_i f_i(s) - g(s) \quad (6)$$

and minimizing the integral of the residual magnitude over the boundary in the least-squares sense. The minimization is with respect to each of the unknown coefficients  $\alpha_i$ :

$$\begin{aligned} \frac{\partial}{\partial \alpha_i^*} \int_s |R(s)|^2 ds &= 0 = \frac{\partial}{\partial \alpha_i^*} \left\{ \sum_{i,j} \alpha_i^* \alpha_j \int_s (L f_i(s))^* L f_j(s) ds \right. \\ &\quad \left. - 2 \operatorname{Re} \sum_i \alpha_i^* \int_s (L f_i(s))^* g(s) ds + \int_s g^*(s) g(s) ds \right\} \\ &= 2 \sum_j \alpha_j \int_s (L f_i(s))^* L f_j(s) ds \\ &\quad - 2 \int_s (L f_i(s))^* g(s) ds \end{aligned} \quad (7)$$

which implies Eq. 5. For point matching, the weighting function is a delta function given by

$$W_i(s) = \delta(s - s_j) \quad (8)$$

so that Eq. 3 becomes

$$\begin{aligned}\bar{M}_{ij} &= \int_s \delta(s - s_j) L f_j(s) ds \\ &= L f_j(s_i)\end{aligned}\quad (9)$$

where  $s_i$  is a sample point along the boundary.

Now consider the practical implementation of Eqs. 2-4. The integrals are usually evaluated numerically so that Eqs. 3 and 4 become sums:

$$\bar{M}_{ij} = \sum_{p=1}^m q_p w_i(s_p) L f_j(s_p) \quad (10)$$

$$g_i = \sum_{p=1}^m q_p w_i(s_p) g(s_p) \quad (11)$$

where  $q_p$  are the weights of a Gaussian quadrature integration method [5].  $m$  is the number of points of the integration method. Assume that the number of functions  $f_i$  in Eq. 1 ranges from 1 ...  $n$ . The matrix equation to solve becomes

$$\begin{bmatrix} \sum_{p=1}^m q_p w_1(s_p) L f_1(s_p) & \cdots & \sum_{p=1}^m q_p w_1(s_p) L f_n(s_p) \\ \vdots & \vdots & \vdots \\ \sum_{p=1}^m q_p w_n(s_p) L f_1(s_p) & \cdots & \sum_{p=1}^m q_p w_n(s_p) L f_n(s_p) \end{bmatrix} \vec{\alpha} = \begin{bmatrix} \sum_{p=1}^m q_p w_1(s_p) g(s_p) \\ \vdots \\ \sum_{p=1}^m q_p w_n(s_p) g(s_p) \end{bmatrix} \quad (12)$$

It can be verified through direct matrix multiplication that Eq. 12 is equivalent to

$$\bar{Q}^t \bar{P} \vec{\alpha} = \bar{Q}^t \vec{G} \quad (13)$$

where  $t$  denotes the matrix transpose and

$$\bar{Q} = \begin{bmatrix} \sqrt{q_1} w_1(s_1) & \cdots & \sqrt{q_1} w_n(s_1) \\ \vdots & \vdots & \vdots \\ \sqrt{q_m} w_1(s_m) & \cdots & \sqrt{q_m} w_n(s_m) \end{bmatrix} \quad (14)$$

$$\bar{P} = \begin{bmatrix} \sqrt{q_1} Lf_1(s_1) & \cdots & \sqrt{q_1} Lf_n(s_1) \\ \vdots & \vdots & \vdots \\ \sqrt{q_m} Lf_1(s_m) & \cdots & \sqrt{q_m} Lf_n(s_m) \end{bmatrix} \quad (15)$$

$$\vec{G} = \begin{bmatrix} \sqrt{q_1} g(s_1) \\ \vdots \\ \sqrt{q_m} g(s_m) \end{bmatrix} \quad (16)$$

The number of rows of the matrix  $\bar{Q}$  is  $m$ , whereas the number of columns is  $n$ . If  $m$  is set equal to  $n$ , then the matrices in Eq. 13 become square; the problem then becomes equivalent to

$$\bar{P} \vec{\alpha} = \vec{G} \quad (17)$$

This equation is equivalent to the point-matching method applied to Eq. 1 in which the number of functions  $f_i$  equals the number of boundary-sampling points. Because this solution no longer depends on the form of the weighting function, it is also equivalent to the boundary residual solution. Now, the boundary-residual method [2, 3] and the method of moments [1] are rigorously convergent, whereas point matching has been shown to fail to converge to the proper solution [6] in some cases. The discrepancy lies in the discretization inherent in the numerical integration routine used to compute Eqs. 3 and 4. By using too few integration points, to where the number of integration sample points ( $m$ ) equals the number of fitting functions ( $n$ ), the method of moments degrades to point matching. This conclusion was also reached by Djordjevic and Sarkar [15] although they do not discuss the failure of point matching as in the next two sections.

# THE FAILURE OF POINT MATCHING WITH FUNCTIONS OF UNBOUNDED VARIATION

Why does point matching fail? Return to the first operation imposed by the method of moments on Eq. 1, i.e., integrating with respect to a weighting function:

$$\int_s W_j(s) \sum_i \alpha_i L f_i(s) ds = \int_s W_j(s) g(s) ds \quad j = 1, \dots, n \quad (18)$$

It is assumed that the integral and the summation may be interchanged in order to create Eq. 2. Titchmarsh [7] proves that an infinite series may be multiplied by a function of bounded variation and integrated term by term. This theorem applies even when the series diverges. Now, the weighting function of the boundary residual method (Eq. 5) is such a function of bounded variation, and so the resulting equations created are valid. The point-matching method uses a delta function as a weighting function (Eq. 8), which is not of bounded variation; bringing the integral inside the summation is not proven to be valid unless the series is uniformly convergent [8], and thus the resulting point-matching equations may or may not be valid. What can be said is that when the series in Eq. 18 satisfies the Rayleigh hypothesis [9], the series converges uniformly [9], and point matching is valid. This view is consistent with Lewin [10].

## THE FAILURE OF POINT MATCHING BY REPEATED LIMITS

This paper shows that point matching, and indeed the method of moments and the boundary-residual method, may fail for another reason. Consider the numerical form of Eq. 18:

$$\sum_{p=1}^m q_p W_j(s_p) \sum_{i=1}^n \alpha_i L f_i(s_p) = \sum_{p=1}^m q_p W_j(s_p) g(s_p) \quad j = 1, \dots, n \quad (19)$$

The matrix form of Eq. 12 implies that

$$\lim_{m \rightarrow \infty} \sum_{p=1}^m q_p W_j(s_p) \lim_{n \rightarrow \infty} \sum_{i=1}^n \alpha_i L f_i(s_p) = \lim_{n \rightarrow \infty} \sum_{i=1}^n \alpha_i \lim_{m \rightarrow \infty} \sum_{p=1}^m q_p W_j(s_p) L f_i(s_p) \quad (20)$$

in order for proper convergence to hold. If the number of integration points ( $m$ ) is large enough, the series form of Eq. 19 will closely approximate the integral form (Eq. 18), and the interchange of series limits should remain valid. Point-matching forces  $m = n$ , and for it to be valid,

the simultaneous double limit ( $m, n \rightarrow \infty$ ) must be a valid operation. This validity does not, in general, hold, as shown by a simple example discussed by Carslaw [11]:

$$s_N(x) = \sum_{p=1}^N f_p(x) \quad (21)$$

where

$$f_p(x) = \frac{1}{(p-1)x+1} - \frac{1}{px+1} \quad (22)$$

$$s_N(x) = 1 - \frac{1}{Nx+1} \quad (23)$$

From Eq. 22, at  $x = 0$ ,

$$f_p(0) = 0 \quad (24)$$

Thus,

$$\lim_{N \rightarrow \infty} (s_N(0)) = s_{\infty}(0) = 0 \quad (25)$$

From Eq. 23, for  $x > 0$ ,

$$\lim_{N \rightarrow \infty} (s_N(x)) = s_{\infty}(x) = 1 \quad x > 0 \quad (26)$$

Thus, the infinite series  $s_{\infty}(x)$  has a discontinuity at  $x = 0$ . It is interesting to note that the partial sum defined by Eq. 21 is a sum of continuous functions, and this is also continuous. The limiting sum  $s_{\infty}$  is not continuous, however, and it is this difference that can cause problems with taking repeated limits.

Consider the limit,

$$\lim_{N \rightarrow \infty} \lim_{x \rightarrow 0} (s_N(x)) = A \quad (27)$$

Let

$$x = \frac{c}{N} \quad (28)$$

where  $c$  is any positive constant. As  $N$  approaches infinity,  $x$  will approach zero, and it seems reasonable that

$$A = \lim_{N \rightarrow \infty} s_n \frac{c}{N} \quad (29)$$

From Eq. 23,

$$A = 1 - \frac{1}{c + 1} \quad (30)$$

Now, for any  $c > 0$ ,  $A$  can be forced to take on any value between 0 and 1 through the proper choice of the constant  $c$ . Thus, the substitution given by Eq. 28 is invalid. It is improper to take a repeated limit of a series in this manner.

It is also improper to exchange the order of the limits in Eq. 27 for, in one case,  $A = 1$ , and in the other,  $A = 0$ , so that

$$\lim_{N \rightarrow \infty} \lim_{x \rightarrow 0} (s_N(x)) \neq \lim_{x \rightarrow 0} \lim_{N \rightarrow \infty} (s_N(x)) \quad (31)$$

The failure of Eq. 31 to be valid is due to the nonuniform convergence of the series for  $x \geq 0$ . That the series defined by Eqs. 21 and 22 is nonuniformly convergent can be seen by considering any  $x$  arbitrarily close to zero. For any arbitrarily small positive number  $\epsilon$ ,

$$|s_\infty(0) - s_N(x)| = \left| \frac{1}{Nx + 1} \right| \leq \epsilon \quad (32)$$

it must be true that

$$N > \frac{\frac{1}{\epsilon} - 1}{x} \quad (33)$$

As  $x$  approaches zero,  $N$  must become large to satisfy Eq. 32.  $N$  must not be dependent on the position within the interval for uniform convergence to hold.

Again, for point matching, the conclusion drawn from this discussion is that point matching is only rigorously valid when the summation in Eq. 1 and Eq. 19 converges uniformly everywhere it is used; satisfaction of the



Rayleigh hypothesis ensures this condition [9]. Unfortunately, determining when a boundary satisfies the Rayleigh hypothesis is not always simple; even a boundary that satisfies the hypothesis can fail through a simple coordinate transformation [10]. Bates [9] suggests using conformal transformations to determine if a boundary satisfies the Rayleigh hypothesis, but this method weakens the main advantage of point matching; i.e., namely simplicity.

Moreover, the implication for the method of moments and the boundary-residual method is that the number of integration points ( $m$ ) must be much larger than the number of functions  $f_i(n)$ . Somewhere between this condition ( $m \gg n$ ) and that of point matching ( $m = n$ ), both of these methods may fail. Indeed, this view is borne out by results found from Ikuno and Yasuura [6] in which their "improved point-matching method" converges for  $m > 2n$ , but fails otherwise.

As a final heuristic argument explaining the failure of point matching, consider a "function fitting" view of this method in which a set of functions ( $Lf_i$  in Eq. 1) is used to fit a driving function ( $g$  in Eq. 1) over an interval (the boundary  $s$ ):

$$\sum_{i=1}^n \alpha_i h_i(s) = g(s) \quad (34)$$

where  $h_i(s) = L f_i(s)$ . Point matching forces this equation to be true on a discrete set of  $n$  points along  $s$ . In between these points, however, the functions  $h_i$  are unconstrained and can take on any value. The measure of the residual of the problem (i.e., how well the fitting functions fit the driving function) is over a discrete set of points of  $g(s)$ , and it is therefore over a set of measure zero on  $g(s)$ . An infinite number of functions can be found which equal  $g(s)$  on a set of measure zero and produce the same point matched solution, even as the number of fitting points ( $n$  in Eq. 1) approaches infinity [2]! The method of moments and the boundary-residual method do not fail because the fitting functions are smoothed everywhere along the boundary by the integral in Eqs. 3 and 4. The residual is not over a set of measure zero, and the fitting functions converge in the mean to the proper value [7].

An example will illustrate this view. Consider a set of odd polynomials used to represent  $\sin(2\pi x)$  over the interval  $0 \leq x \leq 1$ :

$$\sum_{n=0}^N a_n (2\pi x)^{2n+1} \approx \sin(2\pi x) \quad 0 \leq x \leq 1 \quad (35)$$

Figures 1-3 compare the errors of this fit for the case of point matching versus the boundary-residual method. The plots clearly show how the boundary-residual method smooths the error across the entire interval. The error of the point-matching method varies wildly between fitting points, even as the number of fitting functions ( $N$  in Eq. 35) increases.

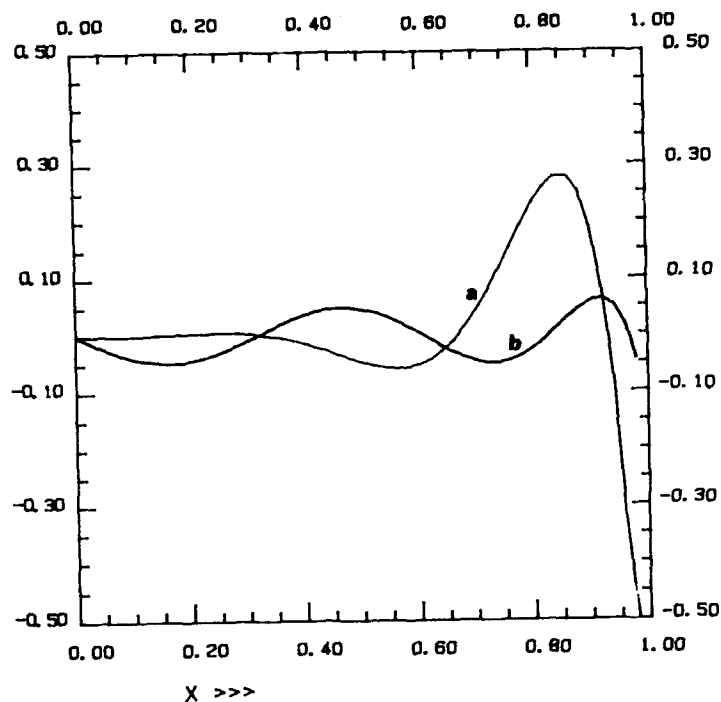


Fig. 1. The errors of Eq. 35 corresponding to the point-matching case (a) and the boundary-residual cases (b) for 4 series terms.

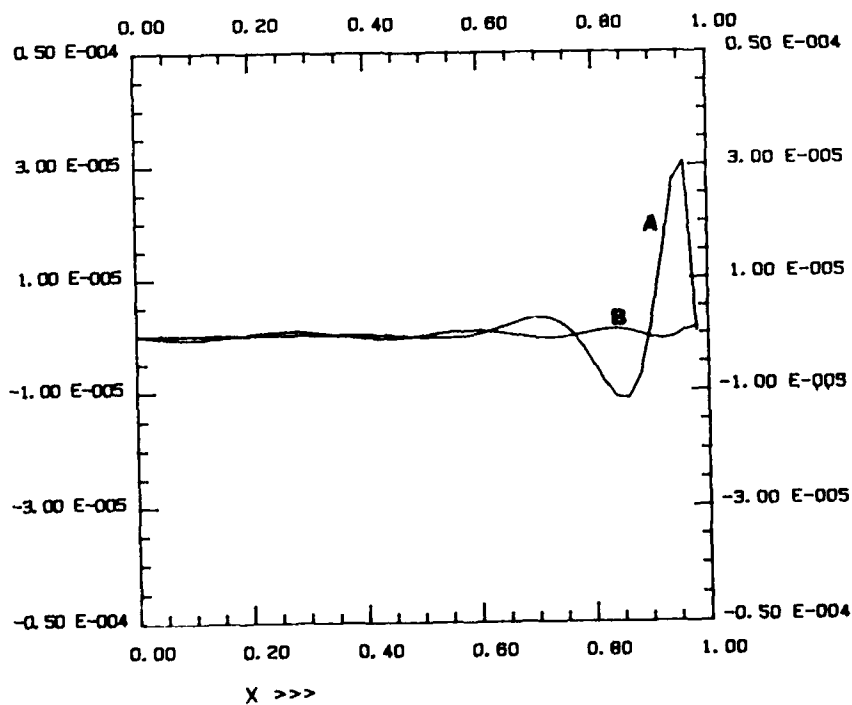


Fig. 2. A comparison of the errors between the point-matched (a) and the boundary-residual solutions (b) for 8 series terms.

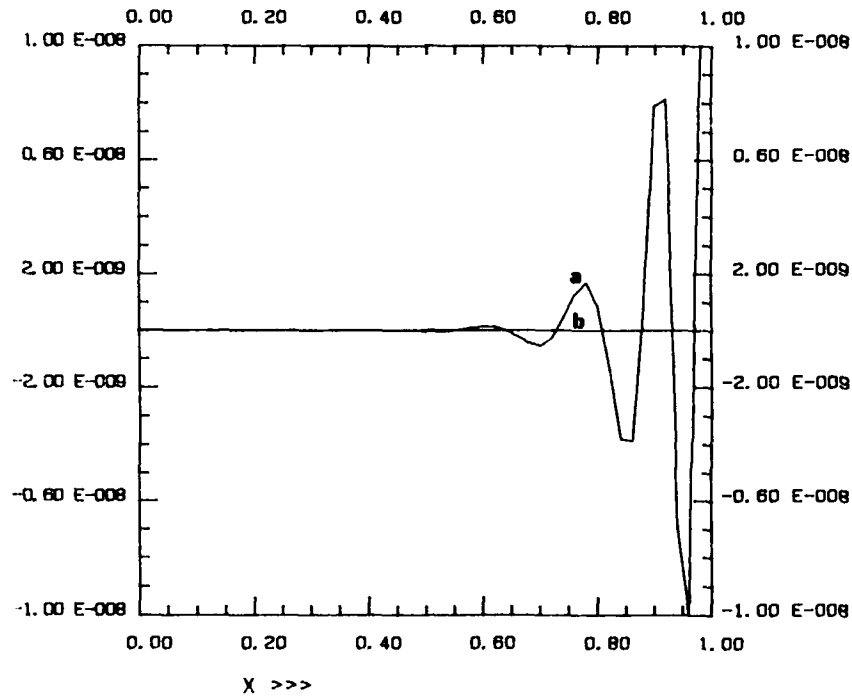


Fig. 3. A comparison of the errors between the point-matched (a) and the boundary-residual solutions (b) for 16 series terms.

#### A REFORMULATION OF POINT MATCHING

Bunch and Grow [12] have proposed a method of using the boundary-residual method which retains most of the simplicity of point matching, but which is rigorously convergent. Recall that in the boundary-residual case, the weighting functions are given by Eq. 5. Using these in Eq. 13 produces

$$\bar{P}^\dagger \bar{P} \vec{\alpha} = \bar{P}^\dagger \vec{G} \quad (36)$$

where  $\dagger$  denotes the complex conjugate transpose, and  $\bar{P}$  is given by Eq. 14. Numerically, this equation is equivalent to solving the equation,

$$\bar{P} \vec{\alpha} \approx \vec{G} \quad (37)$$

in the least-squares sense [12]. Remember,  $m > n$  in Eq. 37, and so there are more rows than columns. Rather than calculating the matrix product in Eq. 36, however, Eq. 37 can be solved directly and equivalently using Householder transforms [12, 13] or using a singular value decomposition [12, 13]. This method retains the advantages of point matching in which a

wave expansion is set up as in Eq. 1 and forced to satisfy the boundary conditions, yet it retains the convergent properties of the boundary-residual method [2, 3]. The method is rigorously convergent because the boundary-residual method (Eq. 36) is rigorously convergent [2, 3], and it creates an identical numerical solution [12, 13] without having to form the matrix product. It is similar to the method proposed by Ikuno and Yasuura [6], except the connection to the method of moments and the boundary-residual method have been clearly shown.

The direct formulation of Eq. 37 also has the advantages of being more numerically stable and quicker to solve than Eq. 36 [12, 13]. The stability problems occur when one or more eigenvalues of the matrix product  $\bar{P}^{\dagger} \bar{P}$  are close to zero, in which case  $\bar{P}^{\dagger} \bar{P}$  is nearly singular. It is easy to show that the eigenvalues of the product  $\bar{P}^{\dagger} \bar{P}$  are the square of the singular values of the matrix  $\bar{P}$ . The matrix  $\bar{P}$  can be decomposed into its singular values,

$$\bar{P} = \bar{V} \bar{\sigma} \bar{U} \quad (38)$$

where  $\bar{\sigma}$  is a diagonal matrix of singular values;  $\bar{V}$  and  $\bar{U}$  are orthogonal matrices in which  $\bar{V}^{\dagger} \bar{V} = \bar{I}$  ( $\bar{U}^{\dagger} \bar{U} = \bar{I}$ ), where  $\bar{I}$  is the identity matrix. Forming the product  $\bar{P}^{\dagger} \bar{P}$ ,

$$\begin{aligned} \bar{P}^{\dagger} \bar{P} &= (\bar{V} \bar{\sigma} \bar{U})^{\dagger} (\bar{V} \bar{\sigma} \bar{U}) \\ &= \bar{U}^{\dagger} (\bar{\sigma})^2 \bar{U} \end{aligned} \quad (39)$$

Thus, if the matrix  $\bar{P}$  has a singular value  $\sigma$  close to zero, the matrix product  $\bar{P}^{\dagger} \bar{P}$  will have a corresponding eigenvalue  $\sigma^2$  even closer to zero. The equation defined by Eq. 36 will thus be more unstable numerically than Eq. 37. Further, solving Eq. 37 directly using a singular value decomposition has the added advantage that the singular values causing numerical instabilities may be discarded in computing the solution [14].

Solving the direct form of the electromagnetic problem may have advantages over using the method of moments. The method of moments creates a matrix problem as in Eq. 12. The matrix consists of a sum for each element due to the numerical integration of the weighting function. The computation of the element sums can be time consuming as the number of sums increases as the square of the matrix size. On the other hand, the direct formulation does not need sums to be computed, but it solves the problem directly. This advantage in speed, however, may be offset by the need for extra storage, as the matrix in the direct formulation is overdetermined (the number of rows is greater than the number of columns). Ikuno and Yasuura [6] have reported good results in a similar formulation when the

numbers of rows (corresponding to boundary points) is greater than twice the number of columns (corresponding to wave expansion functions).

Another consideration is that the singular value decomposition is well-behaved and well-suited for solving the direct formulation in a least-squares sense [13]. The singular value decomposition method allows one to have control over the singular values to produce a well-behaved solution even with a nearly singular set of equations [14]. This control is advantageous when the formulation of the problem produces a nearly singular set of equations as when using a large number of wave expansion functions.

The direct method may also be used for solving scattering problems when the induced current on the scattering surface is expanded as a sum of unknown basis functions. Butler and Wilton [16] have investigated the application of the method of moments as applied to thin-wire scatterers with several different basis sets to represent the wire current. They found the convergence of the solution depended strongly on the basis functions used as well as whether the equations solved were cast in Pocklington (electric field) or Hallén (magnetic vector potential) integral form. Their testing functions were delta functions forcing their method of solution to be that of point matching. As stated, point matching may fail to converge to the correct solution; in this case, point matching was satisfactory because the geometry of the scatterer was simple (the Rayleigh hypothesis was satisfied) and no singularities in the fields existed on the scatterer. Using the direct formulation in this case, however, would allow the technique to be extended to scatterers of more complicated geometry. The dependence of convergence on the choice of basis functions used to represent the wire current would still remain, but an advantage of the direct method is that the singular value decomposition would be ideal for the problem of ill-conditioned matrices found in some of their test cases.

#### A SPHERICAL CAVITY EXAMPLE

To illustrate the use of Eq. 37, we solved the resonances of the spherical cavity using cylindrical wave functions. A scalar expansion for the fields is given by [17]

$$\psi = \sum_n a_n J_\ell(\gamma_n \rho) e^{j\ell\phi} e^{\frac{j2\pi n}{p} z} \quad (40)$$

with

$$\gamma_n = \sqrt{k^2 - \left(\frac{2\pi n}{p}\right)^2} \quad (41)$$

$J_\ell$  is the cylindrical Bessel function of the first kind of order  $\ell$  [17],  $k$  is the wave number ( $\omega/c$ ), and  $p$  is the diameter of the cavity.

This wave expansion is used to create the electric field [17] whose tangential value is minimized on the spherical boundary.

Figure 4 shows the minimum singular value over the wave number of the overdetermined matrix using Eq. 40. In this case, we do not have an incident field and so the right-hand side of Eq. 37 is zero. The minimum singular value of the matrix of Eq. 37 gives an indication of how well the wave expansion (Eq. 40) fits the boundary conditions over frequency [18]. The dips in the singular value are the resonances of the cavity, and these gradually approach the exact resonances (shown as dotted lines) as the number of wave functions ( $n$  in Eq. 67) increases. As shown, good results are obtained using only a few number of wave functions.

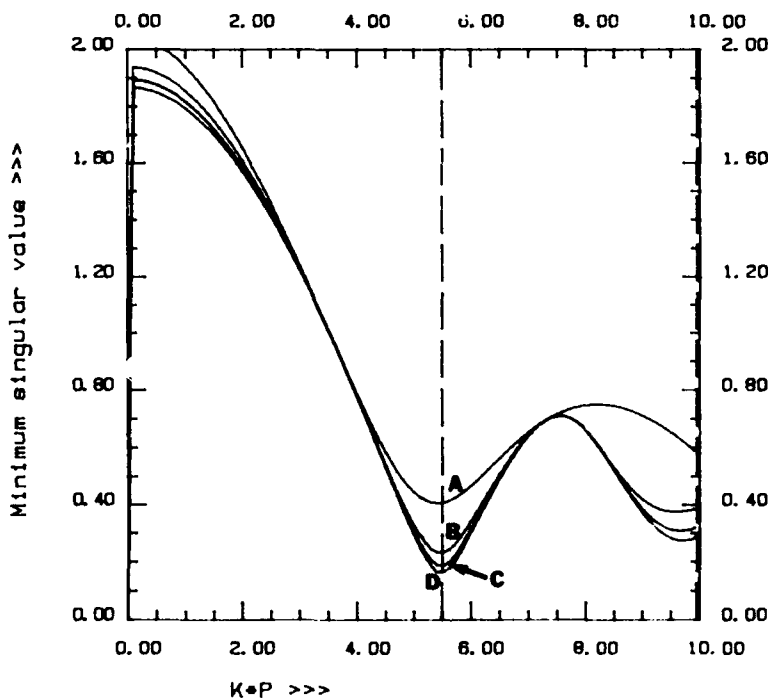


Fig. 4. The minimum singular value over wave number for the spherical cavity with  $\ell = 0$ . The dotted line shows an exact resonance. The results are shown for  $n = 0$  (a),  $n = -1, \dots, 1$  (b),  $-2, \dots, 2$  (c), and  $-3, \dots, 3$  (d).

## CONCLUSIONS

This paper has shown how the method of moments can collapse to the point-matching method and the boundary-residual method; it can do so in two ways. The boundary-residual method has also been shown to revert to point matching in some cases. A large number of sampling points for numerical integration in either the method of moments or the boundary-residual method

can prevent this collapse. The point-matching method is unconstrained between data points; an example has shown that the error of the functions between these points can fluctuate wildly. Finally, a formulation has been given which retains the simplicity of point matching while retaining the rigorous convergence properties of the method of moments or the boundary-residual method.

#### REFERENCES

1. R. Harrington, "Origin and Development of the Method of Moments for Field Computation," IEEE Antennas and Propagation Magazine, Vol. 32, No. 3, June 1990, pp. 31-36.
2. K. J. Bunch, Theoretical and Numerical Foundations of a Boundary-Residual Method for Solving Three-Dimensional Boundary-Value Problems in Electromagnetics, Ph.D. Dissertation, Department of Electrical Engineering, University of Utah, Salt Lake City, Utah, March 1990.
3. J. B. Davies, "A Least-Squares Boundary-Residual Method for the Numerical Solution of Scattering Problems," IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-21, No. 2, February 1973, pp. 99-103.
4. R. H. T. Bates, "The Point-Matching Method for Interior and Exterior Two-Dimensional Boundary-Value Problems," IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-15, March 1967, pp. 185-187.
5. M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions, Dover Publications, New York, 1972, Chapter 25.
6. H. Ikuno and K. Yasuura, "Improved Point-Matching Method with Application to Scattering from a Periodic Surface," IEEE Transactions on Antennas and Propagation, Vol. AP-21, No. 5, September 1973, pp. 657-662.
7. F. S. Titchmarsh, The Theory of Functions, Second Edition, Oxford University Press, London, England, 1939, pp. 421-422.
8. K. Knopp, Theory of Functions Part I, Dover Publications, New York, 1945, Chapter 6.
9. R. H. T. Bates, "Analytic Constraints on Electromagnetic Field Computations," IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-23, No. 8, August 1975, pp. 605-623.
10. L. Lewin, "On the Restricted Validity of Point-Matching Techniques," IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-18, No. 12, December 1970, pp. 1041-1047.

11. H. S. Carslaw, An Introduction to the Theory of Fourier's Series and Integrals, Third Edition, Dover Publications, New York, 1950, Chapter 5.
12. K. J. Bunch and R. W. Grow, "Numerical Aspects of the Boundary-Residual Method," International Journal of Numerical Modeling, Vol. 3, No. 1, March 1990, pp. 57-71.
13. C. L. Lawson and R. J. Hanson, Solving Least-Squares Problems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974, Chapters 3 and 4.
14. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes, Cambridge University Press, Cambridge, 1986, Section 2.9.
15. A. R. Djordjevic and T. K. Sarkar, "A Theorem on the Moment Methods," IEEE Transactions on Antennas and Propagation, Vol. AP-35, No. 3, March 1987, pp. 353-355.
16. C. M. Butler and D. R. Wilton, "Analysis of Various Numerical Techniques Applied to Thin-Wire Scatterers," IEEE Transactions on Antennas and Propagation, July 1975, pp. 534-540.
17. R. F. Harrington, Time-Harmonic Electromagnetic Fields, McGraw-Hill Book Company, New York, 1961, Chapter 5.
18. K. J. Bunch and R. W. Grow, "Numerical Aspects of the Boundary Residual Method," International Journal of Numerical Modeling, Vol. 3, No. 1, March 1990, pp. 57-71.



SPECIAL SECTION  
ON  
TEAM BENCHMARK  
PROBLEM SOLUTIONS

peer-reviewed results of code  
and technique validation efforts

# SOLUTION OF TEAM BENCHMARK PROBLEM #10 (STEEL PLATES AROUND A COIL)

O. Biro

Graz University of Technology, Kopernikusgasse 24, A-8010 Graz, Austria

**Abstract** - Problem No. 10 of the TEAM Workshops is solved by three different finite-element formulations using a magnetic vector potential with the Coulomb gauge and an electric scalar potential. Allowing the normal component of the vector potential to jump at iron/air interfaces yields results in good agreement with measurement data.

## Problem definition

This three-dimensional, non-linear, transient eddy current problem has been proposed by Prof. T. Nakata, N. Takahashi and K. Fujiwara as a benchmark problem for the TEAM Workshops. For convenience, its definition is repeated here [1].

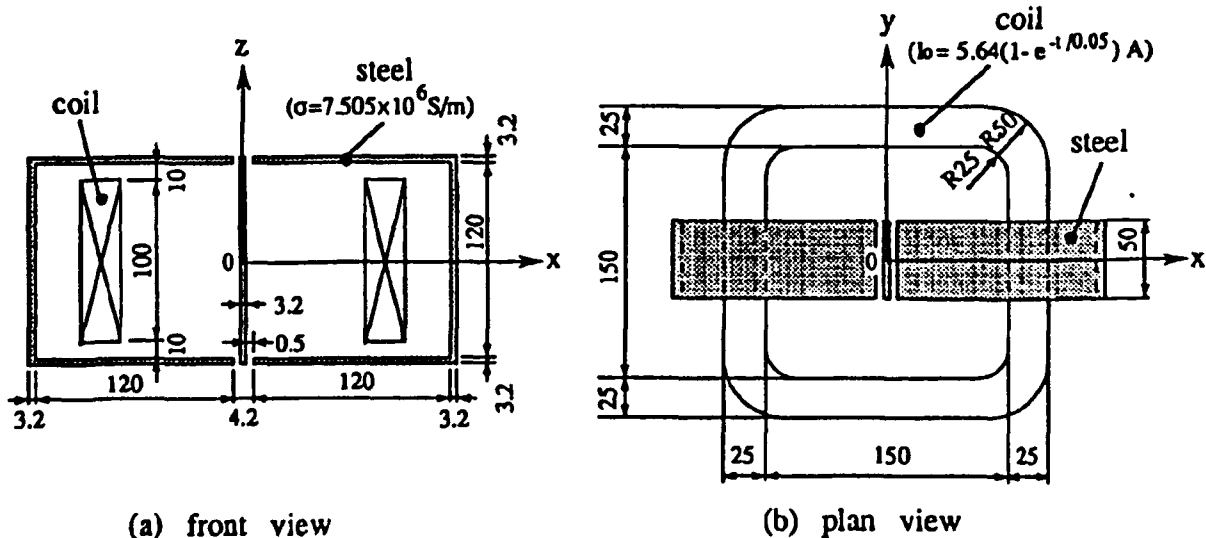


Fig. 1: Steel plates around a coil (dimensions in mm)

The model is shown in Fig. 1. An exciting coil is placed between two steel channels and a steel plate is inserted between the channels. The material of the steel is nonlinear, the magnetization curve is shown in Fig. 2. The curve can be approximated for high flux densities ( $B > 1.8 \text{ T}$ ) as

$$\left. \begin{aligned} B &= \mu_0 H + (aH^2 + bH + c) & (1.8 \text{ T} \leq B \leq 2.22 \text{ T}) \\ B &= \mu_0 H + M_s & (B \geq 2.22 \text{ T}) \end{aligned} \right\} \quad (1)$$

where  $\mu_0$  is the permeability of free space. The constants  $a$ ,  $b$  and  $c$  are  $-2.381 \times 10^{-10}$ ,  $2.327 \times 10^{-6}$  and  $1.590$ , respectively.  $M_s$  is the saturation magnetization ( $2.16 \text{ T}$ ) of the steel. The conductivity of the channels and of the center plate is  $7.505 \times 10^6 \text{ S/m}$ . The number of turns in the coil is 162. The exciting current varies with time as

$$I = \begin{cases} 0 & (t < 0) \\ I_m(1 - e^{-t/\tau}) & (t \geq 0) \end{cases} \quad (2)$$

The amplitude is  $I_m = 5.64$  A and the time constant is  $\tau = 0.05$  s.

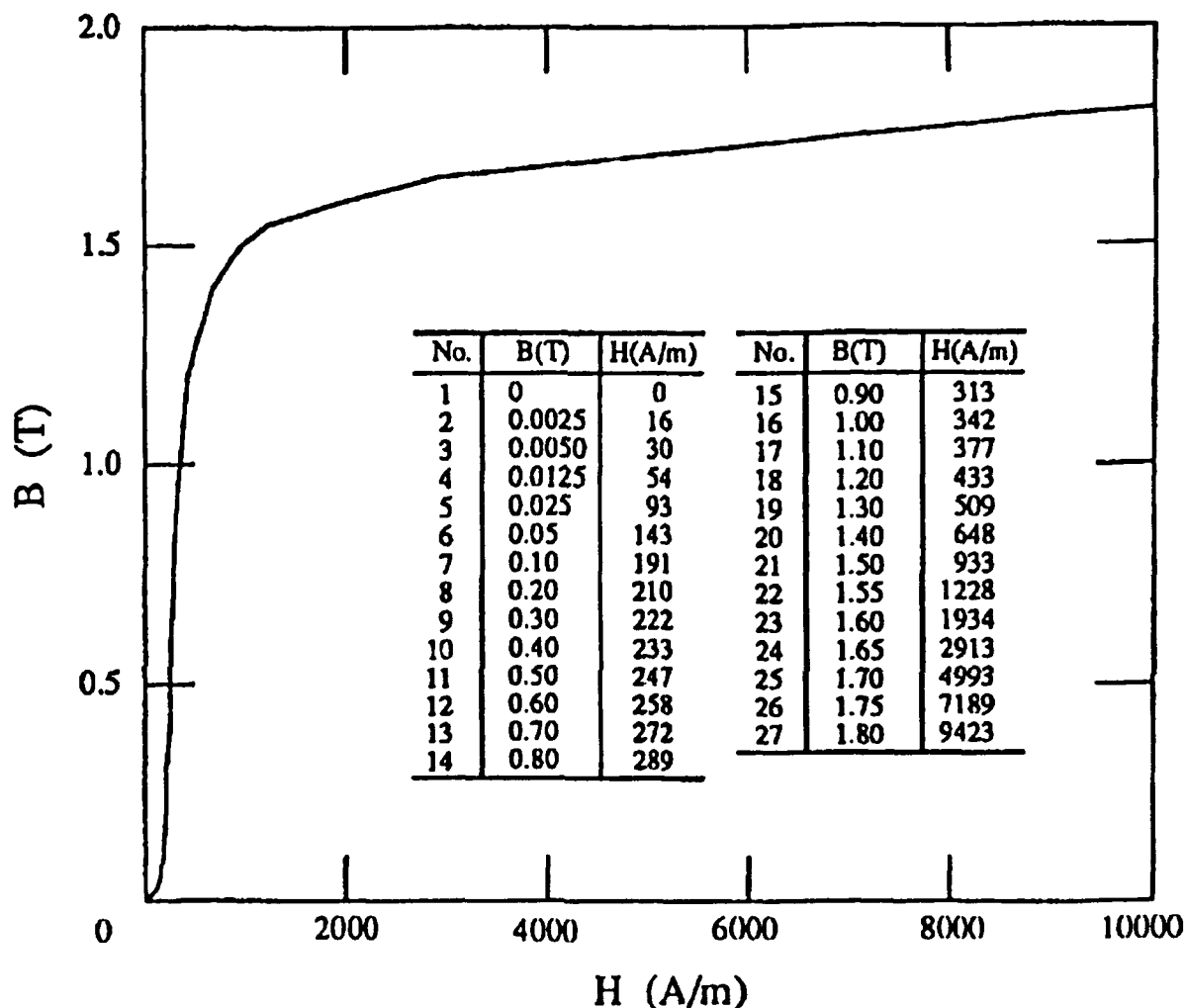


Fig. 2: B-H curve of steel

It is required to find the time functions of the average flux density of the surfaces  $S_1$ ,  $S_2$  and  $S_3$  shown in Fig. 3 and also the time functions of the current density at the points  $P_1$ ,  $P_2$  and  $P_3$ . These quantities have also been measured by the authors of [1].

The problem has been solved with the program package IGTEDDDY of the Institute for Fundamentals and Theory in Electrical Engineering of the Graz University of Technology. Three solutions have been obtained by formulations using a magnetic vector potential throughout and an additional electric scalar potential in the eddy current region.

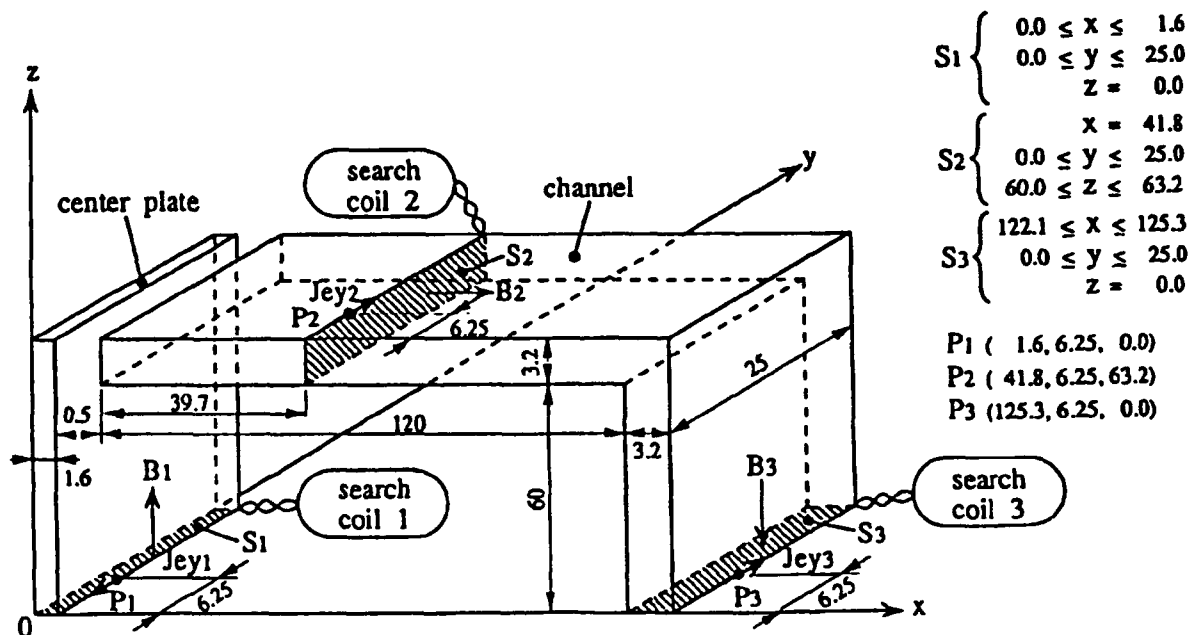


Fig. 3: Measurement positions for flux densities and current densities

#### A, V-A formulation, $A_n$ continuous

This is the well-known A, V-A formulation [2] with the magnetic flux density and the electric field intensity derived from the potentials as

$$\mathbf{B} = \nabla \times \mathbf{A}, \quad (3)$$

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla \frac{\partial v}{\partial t} \quad (4)$$

where  $\mathbf{A}$  is the magnetic vector potential and  $v$  is the time integral of the electric scalar potential. The governing differential equations are

$$\nabla \times (\nu \nabla \times \mathbf{A}) - \nabla (\nu \nabla \cdot \mathbf{A}) + \sigma \frac{\partial \mathbf{A}}{\partial t} + \sigma \nabla \frac{\partial v}{\partial t} = 0 \quad \text{in conductors}, \quad (5)$$

$$\nabla \cdot \left( -\sigma \frac{\partial \mathbf{A}}{\partial t} - \sigma \nabla \frac{\partial v}{\partial t} \right) = 0 \quad \text{in conductors}, \quad (6)$$

$$\nabla \times (\nu \nabla \times \mathbf{A}) - \nabla (\nu \nabla \cdot \mathbf{A}) = \mathbf{J} \quad \text{in non-conductors}. \quad (7)$$

These equations enforce the Coulomb gauge on the vector potential. Using nodal finite elements with one value for each component of  $\mathbf{A}$  in each node, the vector potential is continuous.

The time functions of the average flux density and of the eddy current density in the positions required are plotted in Fig. 4 and in Fig. 5, respectively along with the measured results [1]. The numerical values are shown in Tables 1.1 and 1.2 whereas some further information on the computation is summarized in Table 1.3

Although the discretization is very fine, the average flux density is somewhat lower than measured while the current density is too high.

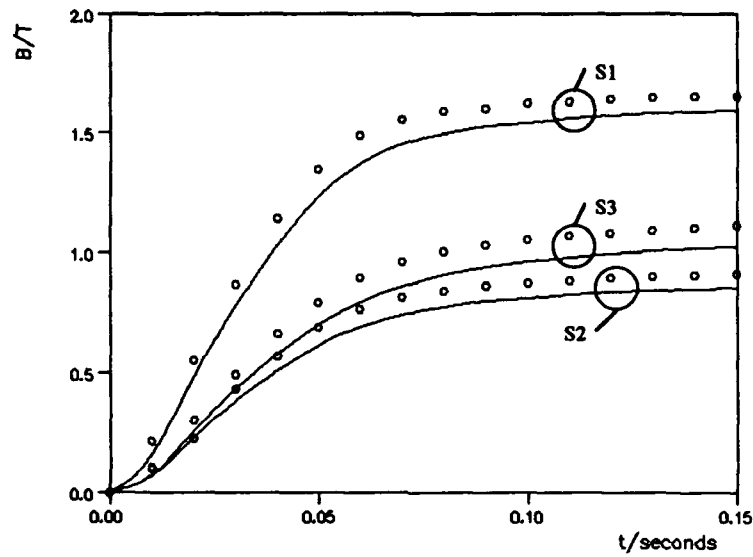


Fig. 4: Time functions of average flux densities, A,V-A formulation,  $A_n$  continuous  
o o o o: measurement, —: computation

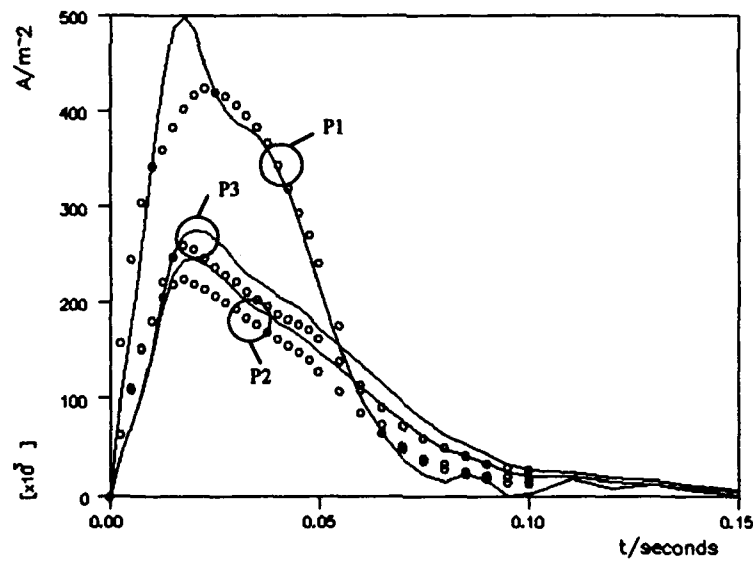


Fig. 5: Time functions of current densities, A,V-A formulation,  $A_n$  continuous  
o o o o: measurement, —: computation

time step	t(s)	position (coordinates in mm)		
		S1 0.0<x<1.6 0.0<y<25.0 z=0.0	S2 x=41.8 0.0<y<25.0 60.0<z<63.2	S3 122.1<x<125.3 0.0<y<25.0 z=0.0
1	0.0025	0.0211	0.0086	0.0090
2	0.0050	0.0548	0.0229	0.0238
3	0.0075	0.1016	0.0431	0.0447
4	0.0100	0.1631	0.0708	0.0741
5	0.0125	0.2375	0.1068	0.1141
6	0.0150	0.3203	0.1479	0.1607
7	0.0175	0.4056	0.1908	0.2098
8	0.0200	0.4894	0.2334	0.2590
9	0.0225	0.5694	0.2745	0.3068
10	0.0250	0.6448	0.3133	0.3521
11	0.0275	0.7163	0.3501	0.3948
12	0.0300	0.7846	0.3848	0.4350
13	0.0325	0.8503	0.4181	0.4733
14	0.0350	0.9139	0.4502	0.5101
15	0.0375	0.9752	0.4811	0.5457
16	0.0400	1.0338	0.5108	0.5798
17	0.0425	1.0896	0.5392	0.6126
18	0.0450	1.1421	0.5662	0.6440
19	0.0475	1.1908	0.5916	0.6737
20	0.0500	1.2356	0.6153	0.7018
21	0.0550	1.3114	0.6566	0.7518
22	0.0600	1.3722	0.6914	0.7951
23	0.0650	1.4191	0.7198	0.8316
24	0.0700	1.4541	0.7426	0.8618
25	0.0750	1.4791	0.7605	0.8866
26	0.0800	1.4970	0.7744	0.9069
27	0.0850	1.5147	0.7876	0.9248
28	0.0900	1.5279	0.7981	0.9398
29	0.0950	1.5339	0.8054	0.9514
30	0.1000	1.5397	0.8118	0.9615
31	0.1100	1.5598	0.8259	0.9807
32	0.1200	1.5713	0.8358	0.9952
33	0.0130	1.5834	0.8447	1.0077
34	0.0140	1.5899	0.8508	1.0168
35	0.0150	1.5928	0.8548	1.0234

Table 1.1: Average flux densities in steel (T)  
A, V-A formulation,  $A_n$  continuous

time step	t(s)	position (coordinates in mm)		
		P1 x=1.6 y=6.25 z=0.0	P2 x=41.8 y=6.25 z=63.2	P3 x=125.3 y=6.25 z=0.0
1	0.0025	1.0648	0.4366	0.4386
2	0.0050	1.7679	0.7297	0.7357
3	0.0075	2.5866	1.0506	1.0598
4	0.0100	3.5277	1.4662	1.5063
5	0.0125	4.3290	1.9602	2.0940
6	0.0150	4.8750	2.2969	2.5105
7	0.0175	4.9736	2.4331	2.7090
8	0.0200	4.8281	2.4520	2.7450
9	0.0225	4.4601	2.3954	2.7396
10	0.0250	4.1797	2.3063	2.6564
11	0.0275	3.9941	2.2157	2.5615
12	0.0300	3.8760	2.1009	2.3980
13	0.0325	3.8194	1.9982	2.2859
14	0.0350	3.7438	1.9240	2.2113
15	0.0375	3.5848	1.8598	2.1302
16	0.0400	3.3809	1.7875	2.0507
17	0.0425	3.1439	1.7260	2.0039
18	0.0450	2.8197	1.6526	1.9295
19	0.0475	2.4771	1.5668	1.8350
20	0.0500	2.1379	1.4761	1.7185
21	0.0550	1.5100	1.3081	1.5442
22	0.0600	0.9941	1.1196	1.3502
23	0.0650	0.6422	0.9297	1.1525
24	0.0700	0.3874	0.7620	0.9494
25	0.0750	0.2213	0.6091	0.7716
26	0.0800	0.1519	0.4821	0.6318
27	0.0850	0.2376	0.4290	0.5311
28	0.0900	0.1697	0.3426	0.4303
29	0.0950	-0.0041	0.2453	0.3208
30	0.1000	0.0295	0.2100	0.2670
31	0.1100	0.1796	0.2109	0.2507
32	0.1200	0.0873	0.1474	0.1882
33	0.0130	0.1158	0.1291	0.1597
34	0.0140	0.0493	0.0871	0.1086
35	0.0150	0.0128	0.0580	0.0737

Table 1.2: Y-component of eddy current  
densities on surface of steel ( $10^5$  A/m<sup>2</sup>)  
A, V-A formulation,  $A_n$  continuous

No	Item	Specification
1	Code name	IGTEDDDY
2	Formulation	FEM (Finite Element Method)
3	Governing equations	$\nabla \times (\nu \nabla \times \mathbf{A}) - \nabla(\nu \nabla \cdot \mathbf{A}) + \sigma \frac{\partial \mathbf{A}}{\partial t} + \sigma \nabla \frac{\partial \nu}{\partial t} = 0 \text{ in conductor}$ $\nabla \times (\nu_0 \nabla \times \mathbf{A}) - \nabla(\nu_0 \nabla \cdot \mathbf{A}) = \mathbf{J} \text{ in vacuum}$
4	Solution variables	$\mathbf{A}, \nu$ in conductor $\mathbf{A}$ in vacuum
5	Gauge condition	imposed on governing equations directly
6	Time difference method	$\theta$ method with $\theta=1$ (backward difference)
7	Technique for non-linear problem	Incremental method
	Convergence criterion	mean $(\Delta \mu_r / \mu_r) < 1\%$ over all Gaussian points max $(\Delta \mu_r / \mu_r) < 5\%$ over all Gaussian points
8	Approximation method of B-H curve	straight lines
9	Technique for open boundary problem	truncation
10	Calculation method of magnetic field produced by exciting current	taking into account exciting current in governing equations directly
11	Property of coefficient matrix of linear equations	symmetric, sparse
12	Solution method for linear equations	ICCG
	Convergence criterion for iteration method	$\ Ax + b\ ^2 / \ b\ ^2 < 10^{-10}$
13	Element type	hexahedron nodal element (20 nodes)
14	Number of elements	7,344
15	Number of nodes	32,986
16	Number of unknowns	88,079
17	Computer	name: DECstation 5000-200 speed: 24 MIPS main memory: 264 MB precision of data: 64 bits CPU time total: 443,117 s

Table 1.3: Computational data, A,V-A formulation,  $A_n$  continuous

### A,V-A formulation, $A_n$ discontinuous

The reason for the above behaviour is that once the Galerkin method is applied to the term  $-\nabla(\nu \nabla \cdot \mathbf{A})$  in the differential equations (5) and (7), the continuity of the quantity  $\nu \nabla \cdot \mathbf{A}$  becomes a natural interface condition [2]. Although  $\nu \nabla \cdot \mathbf{A}$  is zero in the weak sense [2], it does in fact have a nonzero value due to the numerical approximation. This results in a tough constraint on  $\nabla \cdot \mathbf{A}$  along interfaces where the reluctivity  $\nu$  changes abruptly: there must be a jump in the divergence of the vector potential. Thus the accuracy is bound to be poor in the vicinity of such iron/air interfaces. In the present problem with thin ferromagnetic channels, the solution in the entire iron region is bound to be strongly influenced by this inaccuracy.

The problem can be overcome by refining the discretization, so that the condition  $\nu \nabla \cdot \mathbf{A} = 0$  is fulfilled with greater accuracy and the constraint on  $\nabla \cdot \mathbf{A}$  has less effect. Indeed, the experience of the author has shown that a coarser mesh yields much poorer results than those shown above.

The constraint on the continuity of  $\nu \nabla \cdot \mathbf{A}$  can be relaxed by allowing the normal component of the vector potential to be discontinuous on the iron/air interfaces [3]. As a consequence, the natural boundary condition  $\nu \nabla \cdot \mathbf{A} = 0$  results on the interface and the constraint on  $\nabla \cdot \mathbf{A}$  is not present any more. At the application of finite element techniques, the normal component  $A_n$  is allowed to be discontinuous by employing four nodal variables in the nodes on

the interface: the two continuous tangential components and a normal component from the air region as well as one from the iron domain.

The time functions of the average flux density and of the eddy current density obtained by this method in the positions required are plotted in Fig. 6 and in Fig. 7, respectively with the measured results also shown. The numerical values are given in Tables 2.1 and 2.2 whereas some further information on the computation is summarized in Table 2.3

The agreement with the measured results is much better than for the case when  $A_m$  is continuous, although the same mesh has been used. The computation time is somewhat longer, due to the higher number of conjugate gradient iterations needed for the solution of the linear equations systems. It is expected that good results can be obtained by substantially coarser meshes, too.

#### $A_r, V-A_r$ formulation, $A_m$ discontinuous

The finite element mesh used in the above computations does not exactly fit the curved parts of the racetrack coil. This potentially leads to inaccuracies if the total vector potential defined by eq. (3) and the differential equations (5) to (7) are used since the representation of the current density may be inaccurate. To check whether this is the case, a reduced vector potential formulation has also been tried [3]. In this method the magnetic field  $H_s$  and vector potential  $A_s$  due to the coil in free space are split from the solution and it is therefore irrelevant whether the coil is exactly modelled by the finite element mesh.

The potentials are defined by

$$\mathbf{B} = \mu_0 \mathbf{H}_s + \nabla \times \mathbf{A}_r, \quad (8)$$

$$\mathbf{E} = -\frac{\partial \mathbf{A}_s}{\partial t} - \frac{\partial \mathbf{A}_r}{\partial t} - \nabla \frac{\partial v}{\partial t} \quad (9)$$

where  $A_r$  is the reduced vector potential. The governing differential equations are

$$\nabla \times (\nu \nabla \times \mathbf{A}_r) - \nabla (\nu \nabla \cdot \mathbf{A}_r) + \sigma \frac{\partial \mathbf{A}_r}{\partial t} + \sigma \nabla \frac{\partial v}{\partial t} = -\sigma \frac{\partial \mathbf{A}_s}{\partial t} - \nabla \times (\nu \mu_0 \mathbf{H}_s) \quad \text{in conductors,} \quad (10)$$

$$\nabla \cdot \left( -\sigma \frac{\partial \mathbf{A}_r}{\partial t} - \sigma \nabla \frac{\partial v}{\partial t} \right) = \nabla \cdot \left( \sigma \frac{\partial \mathbf{A}_s}{\partial t} \right) \quad \text{in conductors,} \quad (11)$$

$$\nabla \times (\nu_0 \nabla \times \mathbf{A}_r) - \nabla (\nu_0 \nabla \cdot \mathbf{A}_r) = 0 \quad \text{in non-conductors.} \quad (12)$$

In order to avoid the inaccuracies due to the continuity of  $A_m$  in the vicinity of the iron/air interfaces, the normal component of the reduced vector potential has been allowed to be discontinuous here.

The time functions of the average flux density and of the eddy current density obtained by this method in the positions required are plotted in Fig. 8 and in Fig. 9, respectively with the measured results also shown. The numerical values are given in Tables 3.1 and 3.2 whereas some further information on the computation is summarized in Table 3.3.



The results are practically identical with those obtained by the total vector potential, i.e. the inaccurate modelling of the coils has caused no loss of precision in the A,V-A version.

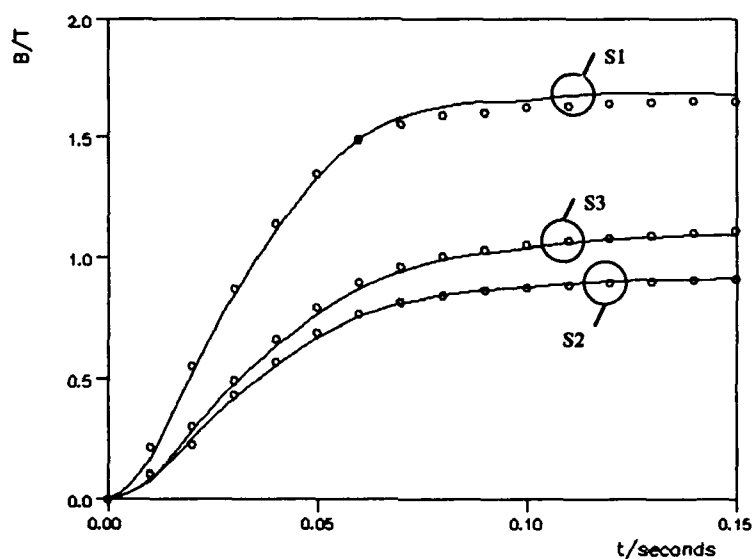


Fig. 6: Time functions of average flux densities, A,V-A formulation,  $A_n$  discontinuous  
o o o o: measurement, —: computation

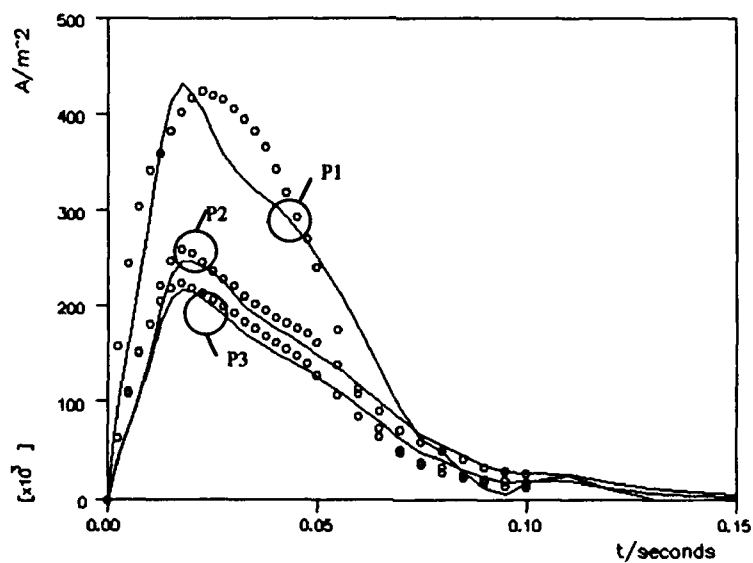


Fig. 7: Time functions of current densities, A,V-A formulation,  $A_n$  discontinuous  
o o o o: measurement, —: computation

time step	t(s)	position (coordinates in mm)		
		S1	S2	S3
		0.0<x<1.6 0.0<y<25.0 z=0.0	x=41.8 0.0<y<25.0 60.0<z<63.2	122.1<x<125.3 0.0<y<25.0 z=0.0
1	0.0025	0.0222	0.0093	0.0098
2	0.0050	0.0582	0.0247	0.0259
3	0.0075	0.1090	0.0469	0.0491
4	0.0100	0.1753	0.0774	0.0818
5	0.0125	0.2564	0.1172	0.1266
6	0.0150	0.3465	0.1624	0.1783
7	0.0175	0.4394	0.2095	0.2325
8	0.0200	0.5300	0.2557	0.2861
9	0.0225	0.6170	0.3004	0.3381
10	0.0250	0.6991	0.3429	0.3878
11	0.0275	0.7765	0.3829	0.4345
12	0.0300	0.8501	0.4206	0.4783
13	0.0325	0.9204	0.4565	0.5198
14	0.0350	0.9879	0.4909	0.5595
15	0.0375	1.0528	0.5239	0.5976
16	0.0400	1.1153	0.5557	0.6343
17	0.0425	1.1751	0.5862	0.6696
18	0.0450	1.2319	0.6153	0.7034
19	0.0475	1.2854	0.6429	0.7356
20	0.0500	1.3355	0.6689	0.7662
21	0.0550	1.4217	0.7150	0.8212
22	0.0600	1.4918	0.7540	0.8687
23	0.0650	1.5457	0.7857	0.9088
24	0.0700	1.5842	0.8105	0.9416
25	0.0750	1.6094	0.8292	0.9677
26	0.0800	1.6298	0.8448	0.9896
27	0.0850	1.6412	0.8562	1.0069
28	0.0900	1.6465	0.8644	1.0203
29	0.0950	1.6488	0.8706	1.0310
30	0.1000	1.6565	0.8778	1.0417
31	0.1100	1.6768	0.8927	1.0620
32	0.1200	1.6858	0.9022	1.0763
33	0.0130	1.6872	0.9075	1.0856
34	0.0140	1.6852	0.9105	1.0916
35	0.0150	1.6826	0.9123	1.0955

Table 2.1: Average flux densities in steel (T)  
A,V-A formulation,  $A_n$  discontinuous

time step	t(s)	position (coordinates in mm)		
		P1	P2	P3
		x=1.6 y=6.25 z=0.0	x=41.8 y=6.25 z=63.2	x=125.3 y=6.25 z=0.0
1	0.0025	1.0263	0.4403	0.4458
2	0.0050	1.6489	0.7311	0.7440
3	0.0075	2.2837	1.0457	1.0698
4	0.0100	2.9981	1.4119	1.4821
5	0.0125	3.6971	1.8179	1.9900
6	0.0150	4.1405	2.0703	2.3195
7	0.0175	4.3106	2.1641	2.4631
8	0.0200	4.2053	2.1471	2.4638
9	0.0225	4.0380	2.0731	2.4047
10	0.0250	3.7800	1.9908	2.3406
11	0.0275	3.5729	1.8946	2.2397
12	0.0300	3.4316	1.7901	2.1117
13	0.0325	3.3226	1.7002	1.9825
14	0.0350	3.2144	1.6262	1.9017
15	0.0375	3.1307	1.5610	1.8203
16	0.0400	3.0437	1.5029	1.7528
17	0.0425	2.9261	1.4455	1.6955
18	0.0450	2.7909	1.3803	1.6219
19	0.0475	2.6483	1.3144	1.5519
20	0.0500	2.5040	1.2477	1.4841
21	0.0550	2.1577	1.1114	1.3464
22	0.0600	1.7497	0.9513	1.1735
23	0.0650	1.3192	0.7846	0.9991
24	0.0700	0.9306	0.6234	0.8244
25	0.0750	0.6056	0.4805	0.6653
26	0.0800	0.4867	0.4056	0.5626
27	0.0850	0.2718	0.3037	0.4467
28	0.0900	0.1198	0.2248	0.3503
29	0.0950	0.0586	0.1741	0.2815
30	0.1000	0.1824	0.1934	0.2735
31	0.1100	0.2435	0.1931	0.2564
32	0.1200	0.1078	0.1252	0.1790
33	0.0130	0.0112	0.0738	0.1192
34	0.0140	-0.0316	0.0429	0.0774
35	0.0150	-0.0252	0.0275	0.0506

Table 2.2: Y-component of eddy current  
densities on surface of steel ( $10^5$  A/m<sup>2</sup>)  
A,V-A formulation,  $A_n$  discontinuous

No	Item	Specification
1	Code name	IGTEDDDY
2	Formulation	FEM (Finite Element Method)
3	Governing equations	$\nabla \times (\nu \nabla \times \mathbf{A}) - \nabla(\nu \nabla \cdot \mathbf{A}) + \sigma \frac{\partial \mathbf{A}}{\partial t} + \sigma \nabla \frac{\partial \phi}{\partial t} = \mathbf{0} \text{ in conductor}$ $\nabla \times (\nu_0 \nabla \times \mathbf{A}) - \nabla(\nu_0 \nabla \cdot \mathbf{A}) = \mathbf{J} \text{ in vacuum}$
4	Solution variables	$\mathbf{A}, \nu$ in conductor $\mathbf{A}$ in vacuum
5	Gauge condition	imposed on governing equations directly, $A_n$ discontinuous on iron/air interface
6	Time difference method	$\theta$ method with $\theta=1$ (backward difference)
7	Technique for non-linear problem	Incremental method
	Convergence criterion	$\text{mean}(\Delta\mu_r / \mu_r) < 1\%$ over all Gaussian points $\text{max}(\Delta\mu_r / \mu_r) < 5\%$ over all Gaussian points
8	Approximation method of B-H curve	straight lines
9	Technique for open boundary problem	truncation
10	Calculation method of magnetic field produced by exciting current	taking into account exciting current in governing equations directly
11	Property of coefficient matrix of linear equations	symmetric, sparse
12	Solution method for linear equations	ICCG
	Convergence criterion for iteration method	$\ Ax + b\ ^2 / \ b\ ^2 < 10^{-10}$
13	Element type	hexahedron nodal element (20 nodes)
14	Number of elements	7,344
15	Number of nodes	32,986
16	Number of unknowns	89,278
17	Computer	name: DECstation 5000-200 speed: 24 MIPS main memory: 264 MB precision of data: 64 bits CPU time total: 663,663 s

Table 2.3: Computational data, A,V-A formulation,  $A_n$  discontinuous

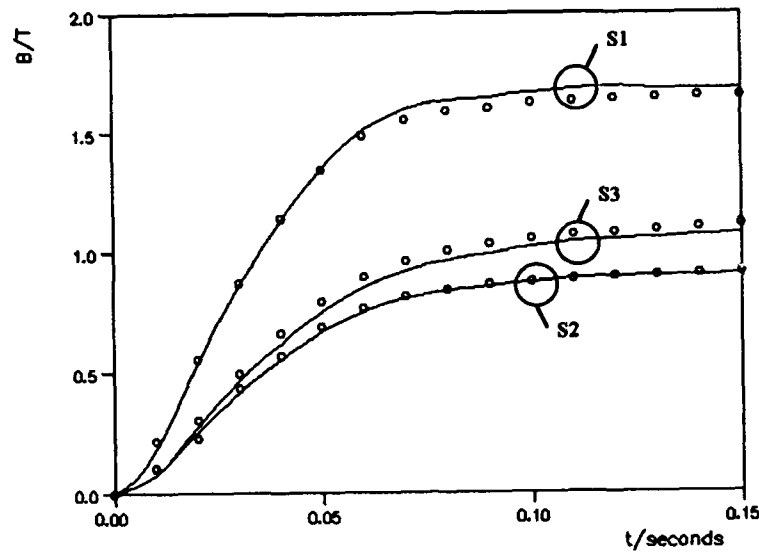


Fig. 8: Time functions of average flux densities,  $A_r$ , V- $A_r$  formulation,  $A_m$  discontinuous  
 o o o o: measurement, ———: computation

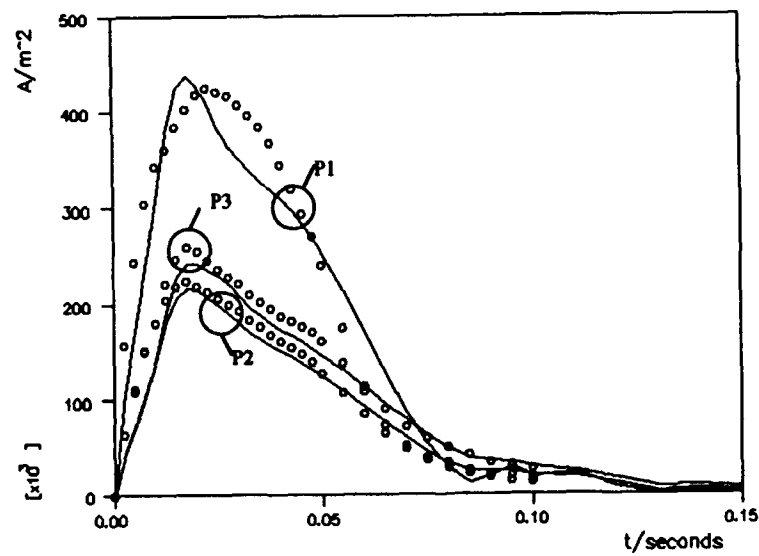


Fig. 9: Time functions of current densities,  $A_r$ , V- $A_r$  formulation,  $A_m$  discontinuous  
 o o o o: measurement, ———: computation

time step	t(s)	position (coordinates in mm)		
		S1 0.0<x<1.6 0.0<y<25.0 z=0.0	S2 x=41.8 0.0<y<25.0 60.0<z<63.2	S3 122.1<x<125.3 0.0<y<25.0 z=0.0
1	0.0025	0.07	0.0093	0.0092
2	0.0050	0.012	0.0249	0.0246
3	0.0075	0.1142	0.0474	0.0470
4	0.0100	0.1824	0.0781	0.0787
5	0.0125	0.2651	0.1179	0.1223
6	0.0150	0.3566	0.1632	0.1731
7	0.0175	0.4508	0.2103	0.2265
8	0.0200	0.5425	0.2565	0.2793
9	0.0225	0.6305	0.3011	0.3304
10	0.0250	0.7134	0.3435	0.3794
11	0.0275	0.7917	0.3836	0.4255
12	0.0300	0.8664	0.4214	0.4688
13	0.0325	0.9377	0.4574	0.5098
14	0.0350	1.0062	0.4918	0.5489
15	0.0375	1.0721	0.5249	0.5865
16	0.0400	1.1354	0.5567	0.6227
17	0.0425	1.1960	0.5872	0.6576
18	0.0450	1.2533	0.6163	0.6909
19	0.0475	1.3071	0.6438	0.7225
20	0.0500	1.3571	0.6696	0.7525
21	0.0550	1.4425	0.7150	0.8061
22	0.0600	1.5110	0.7530	0.8522
23	0.0650	1.5625	0.7835	0.8906
24	0.0700	1.5982	0.8070	0.9216
25	0.0750	1.6200	0.8242	0.9459
26	0.0800	1.6311	0.8364	0.9646
27	0.0850	1.6367	0.8453	0.9794
28	0.0900	1.6453	0.8542	0.9929
29	0.0950	1.6574	0.8636	1.0061
30	0.1000	1.6653	0.8711	1.0171
31	0.1100	1.6842	0.8854	1.0370
32	0.1200	1.6916	0.8942	1.0508
33	0.0130	1.6812	0.8952	1.0568
34	0.0140	1.6826	0.8993	1.0640
35	0.0150	1.6829	0.9024	1.0698

Table 3.1: Average flux densities in steel (T)  
 $A_r$ , V- $A_r$  formulation,  $A_m$  discontinuous

time step	t(s)	position (coordinates in mm)		
		P1 x=1.6 y=6.25 z=0.0	P2 x=41.8 y=6.25 z=63.2	P3 x=125.3 y=6.25 z=0.0
1	0.0025	1.0805	0.4390	0.4155
2	0.0050	1.7310	0.7360	0.7099
3	0.0075	2.3863	1.0575	1.0303
4	0.0100	3.0836	1.4188	1.4316
5	0.0125	3.7678	1.8206	1.9378
6	0.0150	4.2121	2.0724	2.2730
7	0.0175	4.3720	2.1633	2.4171
8	0.0200	4.2573	2.1450	2.4245
9	0.0225	4.0869	2.0678	2.3598
10	0.0250	3.8132	1.9864	2.3002
11	0.0275	3.6167	1.8924	2.2074
12	0.0300	3.4921	1.7916	2.0755
13	0.0325	3.3717	1.7007	1.9641
14	0.0350	3.2707	1.6273	1.8719
15	0.0375	3.1866	1.5630	1.8011
16	0.0400	3.0944	1.5038	1.7303
17	0.0425	2.9617	1.4444	1.6672
18	0.0450	2.8246	1.3779	1.5989
19	0.0475	2.6758	1.3089	1.5261
20	0.0500	2.5130	1.2350	1.4533
21	0.0550	2.1364	1.0942	1.3093
22	0.0600	1.6995	0.9275	1.1349
23	0.0650	1.2626	0.7554	0.9568
24	0.0700	0.8532	0.5922	0.7829
25	0.0750	0.5265	0.4450	0.6216
26	0.0800	0.2665	0.3246	0.4855
27	0.0850	0.1252	0.2455	0.3863
28	0.0900	0.2064	0.2399	0.3515
29	0.0950	0.2913	0.2474	0.3375
30	0.1000	0.1912	0.1981	0.2834
31	0.1100	0.2247	0.1855	0.2504
32	0.1200	0.0862	0.1171	0.1758
33	0.0130	-0.1320	0.0236	0.0809
34	0.0140	0.0220	0.0546	0.0907
35	0.0150	0.0108	0.0423	0.0730

Table 3.2: Y-component of eddy current  
densities on surface of steel ( $10^5$  A/m<sup>2</sup>)  
 $A_r$ , V- $A_r$  formulation,  $A_m$  discontinuous

No	Item	Specification
1	Code name	IGTEDDDY
2	Formulation	FEM (Finite Element Method)
3	Governing equations	$\nabla \times (\nu \nabla \times \mathbf{A}_r) - \nabla (\nu \nabla \cdot \mathbf{A}_r) + \sigma \frac{\partial \mathbf{A}_r}{\partial t} + \sigma \nabla \frac{\partial \psi}{\partial t} =$ $- \sigma \frac{\partial \mathbf{A}_s}{\partial t} - \nabla \times (\nu \mu_0 \mathbf{H}_s) \quad \text{in conductor}$ $\nabla \times (\nu_0 \nabla \times \mathbf{A}_r) - \nabla (\nu_0 \nabla \cdot \mathbf{A}_r) = 0 \quad \text{in vacuum}$
4	Solution variables	$\mathbf{A}_r, \psi$ in conductor $\mathbf{A}_r$ in vacuum
5	Gauge condition	imposed on governing equations directly, $\mathbf{A}_m$ discontinuous on iron/air interface
6	Time difference method	$\theta$ method with $\theta=1$ (backward difference)
7	Technique for non-linear problem	Incremental method
	Convergence criterion	$\text{mean}(\Delta \mu_r / \mu_r) < 1\%$ over all Gaussian points $\text{max}(\Delta \mu_r / \mu_r) < 5\%$ over all Gaussian points
8	Approximation method of B-H curve	straight lines
9	Technique for open boundary problem	truncation
10	Calculation method of magnetic field produced by exciting current	Biot-Savart law (analytical) Biot-Savart law (numerical)
11	Property of coefficient matrix of linear equations	symmetric, sparse
12	Solution method for linear equations	ICCG
	Convergence criterion for iteration method	$\ Ax + b\ ^2 / \ b\ ^2 < 10^{-10}$
13	Element type	hexahedron nodal element (20 nodes)
14	Number of elements	7,344
15	Number of nodes	32,986
16	Number of unknowns	89,278
17	Computer	name: DECstation 5000-200 speed: 24 MIPS main memory: 264 MB precision of data: 64 bits CPU time total: 685,377 s

Table 3.3: Computational data,  $\mathbf{A}_r, \mathbf{V}-\mathbf{A}_r$  formulation,  $\mathbf{A}_m$  discontinuous

## References

- [1] T. Nakata, N. Takahashi and K. Fujiwara, "Summary of results for benchmark problem 10 (steel plates around a coil)", in R. Albanese, E. Coccoresse, Y. Crutzen and P. Molino (ed.), *Proc., Third International TEAM Workshop*, Sorrento, Italy, 12-13 July 1991, pp. 211-221
- [2] O. Biro, K. Preis, "On the use of the magnetic vector potential in the finite element analysis of 3-D eddy currents", *IEEE Transactions on Magnetism*, vol. MAG-25, 1989, pp. 3145-3159
- [3] K. Preis, I. Bardi, O. Biro, C. Magele, W. Renhart, K.R. Richter and G. Vrisk, "Numerical analysis of 3-D magnetostatic fields", *IEEE Transactions on Magnetism*, vol. MAG-27, 1991, pp. 3798-3803

# SOLUTION OF TEAM BENCHMARK PROBLEM #13 (3-D NONLINEAR MAGNETOSTATIC MODEL)

O. Biro, Ch. Magele, G. Vrisk

Graz University of Technology, Kopernikusgasse 24, A-8010 Graz, Austria

**Abstract** - Problem No. 13 of the TEAM Workshops is solved by two scalar potential and one vector potential finite-element formulations. The results obtained by the different scalar potential methods are identical and their agreement with those yielded by the vector potential approach and also with measurement data is satisfactory.

## Problem definition

This three-dimensional, non-linear, magnetostatic problem has been proposed by Prof. T. Nakata and K. Fujiwara as a benchmark problem for the TEAM Workshops. For convenience, its definition is repeated here [1,6].

The model is shown in Fig. 1. An exciting coil is placed between two steel channels shifted as shown and a steel plate is inserted between the channels. The material of the steel is nonlinear, the magnetization curve is shown in Fig. 2. The curve can be approximated for high flux densities ( $B > 1.8 \text{ T}$ ) as

$$\left. \begin{aligned} B &= \mu_0 H + (aH^2 + bH + c) & (1.8 \text{ T} \leq B \leq 2.22 \text{ T}) \\ B &= \mu_0 H + M_s & (B \geq 2.22 \text{ T}) \end{aligned} \right\} \quad (1)$$

where  $\mu_0$  is the permeability of free space. The constants  $a$ ,  $b$  and  $c$  are  $-2.822 \times 10^{-10}$ ,  $2.529 \times 10^{-5}$  and  $1.591$ , respectively.  $M_s$  is the saturation magnetization ( $2.16 \text{ T}$ ) of the steel. The coil is excited by a d.c. current. The total current is  $1000 \text{ AT}$  in one case and  $3000 \text{ AT}$  in the other. Presently the problem is only open for the  $1000 \text{ AT}$  case.

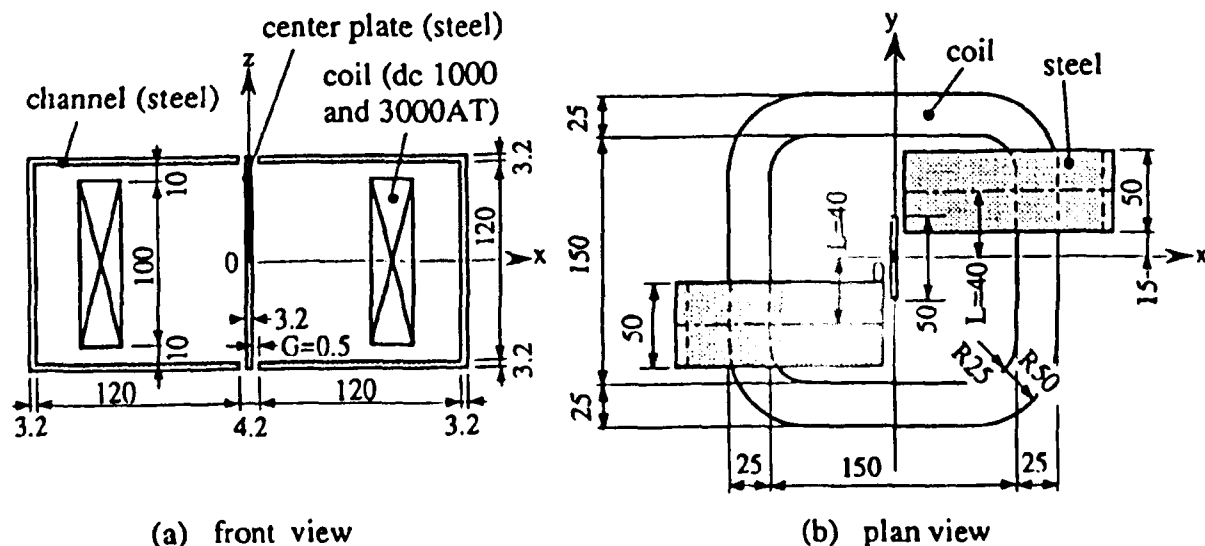


Fig. 1: 3-D nonlinear magnetostatic model (dimensions in mm)

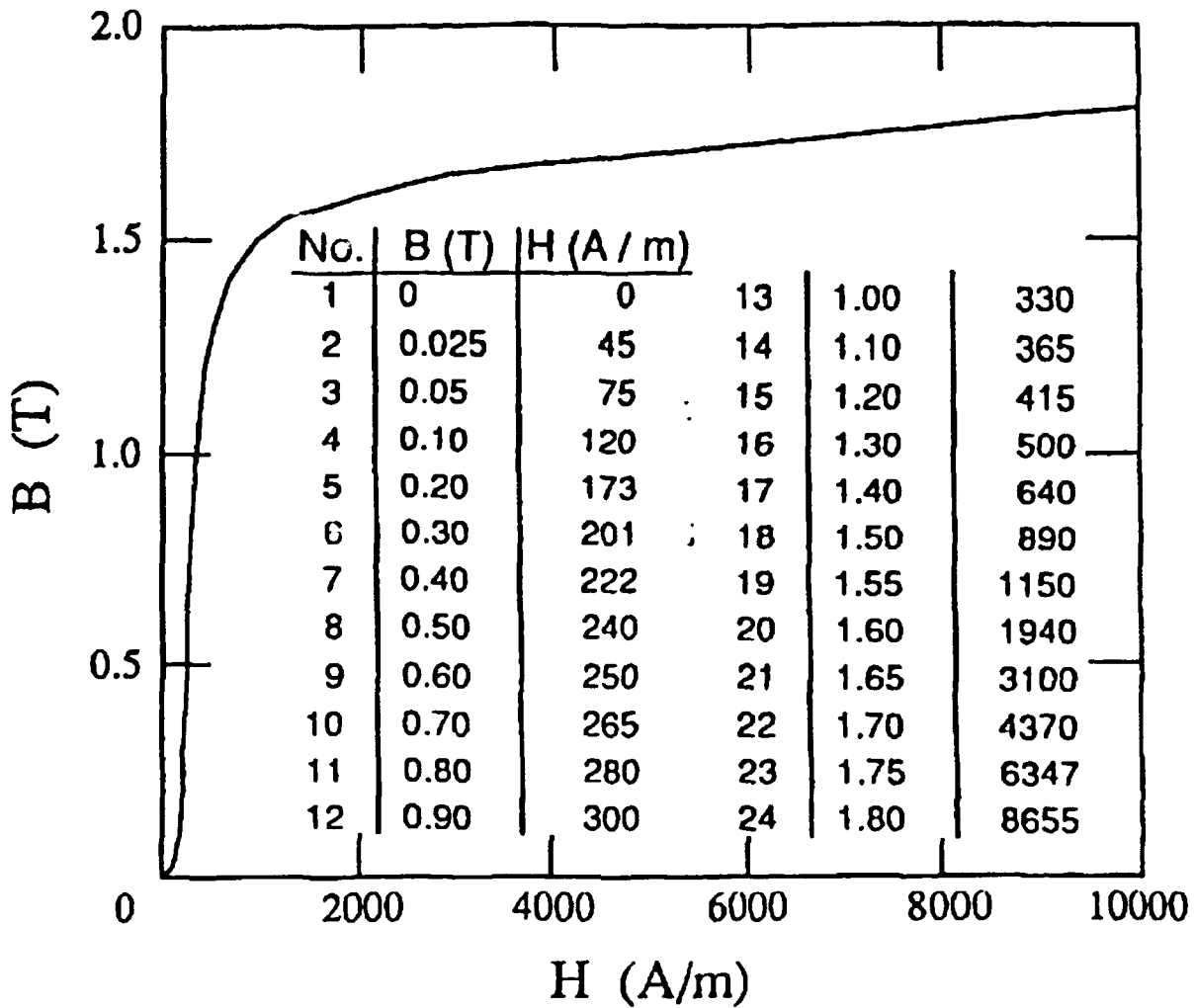


Fig. 2: B-H curve of steel

It is required to obtain the average flux densities at several locations in the channels and in the center plate as well as along a line and at some specified points in air (see e.g. Tables 1.1 to 1.3)

The problem has been solved with the program package IGTEMAG3D of the Institute for Fundamentals and Theory in Electrical Engineering of the Graz University of Technology. Three solutions have been obtained, two by formulations using a magnetic scalar potential and one by employing a magnetic vector potential. The finite element meshes have been selected so that the number of degrees of freedom is about 200,000.

#### $\Phi$ - $\Psi$ formulation

This is the well known formulation in terms of a reduced and a total magnetic scalar potential [2]. The magnetic field intensity in the free space region is written as



$$\mathbf{H} = \mathbf{H}_s - \nabla\Phi \quad (2)$$

using the reduced scalar potential  $\Phi$  and the source field  $\mathbf{H}_s$  due to the coil in free space computed by Biot-Savart integration. In the iron regions, the magnetic field intensity can be derived from the total scalar potential  $\Psi$ :

$$\mathbf{H} = -\nabla\Psi. \quad (3)$$

The two potentials are linked at the interface using the continuity condition of the tangential component of  $\mathbf{H}$ :

$$\Phi = \Psi - \int \mathbf{H}_{st} \cdot d\mathbf{s}. \quad (4)$$

The average flux density values in the three sections of the channel, along the specified line in the air and at the specified points are shown in Tables 1.1, 1.2 and 1.3. Some further information concerning the computation is summarized in Table 1.4.

#### **T- $\Phi$ formulation**

This is the well known  $\mathbf{T}$ - $\Omega$  method [3] where the magnetic field intensity is written as

$$\mathbf{H} = \mathbf{T} - \nabla\Phi. \quad (5)$$

The function  $\mathbf{T}$  is selected to satisfy

$$\nabla \times \mathbf{T} = \mathbf{J}. \quad (6)$$

In the present calculation  $\mathbf{T}$  was chosen to have a single axial component assuming a constant value in the air core of the racetrack coil, linearly decreasing to zero within the windings and zero outside the coil. To avoid cancellation errors,  $\mathbf{T}$  was represented with the aid of edge elements by computing its integral along each edge in the finite element mesh [4].

The average flux density values in the three sections of the channel, along the specified line in the air and at the specified points are shown in Tables 2.1, 2.2 and 2.3. Some further information concerning the computation is summarized in Table 2.4.

The results are practically identical to those obtained by the  $\Phi$ - $\Psi$  formulation. The computation time is somewhat lower since no Biot-Savart integration is necessary. In the conjugate gradient iterations, it suffices to use a convergence criterion of  $10^{-7}$  instead of  $10^{-12}$  with respect to the right hand side vector in order to attain the same precision in the solution.

No.	coordinates (mm)			B(T)
	x	y	z	
1			0.0	1.420
2			10.0	1.406
3			20.0	1.373
4	0.0<x<1.6	-25.0<y<25.0	30.0	1.317
5			40.0	1.232
6			50.0	1.072
7			60.0	0.608
8	2.1			0.320
9	10.0			0.594
10	20.0			0.678
11	30.0			0.735
12	40.0			0.785
13	50.0	15.0<y<65.0	60.0<z<63.2	0.827
14	60.0			0.865
15	80.0			0.931
15	100.0			0.974
17	110.0			0.980
18	122.1			0.950
19			60.0	0.885
20			50.0	0.988
21			40.0	0.994
22	122.1<x<125.3	15.0<y<65.0	30.0	0.999
23			20.0	1.003
24			10.0	1.006
25			0.0	1.007

Table 1.1: Average flux densities in steel (T)  
 $\Phi$ - $\Psi$  formulation

No	coordinates (mm)			B(T)
	x	y	z	
26	10.0			0.0348
27	20.0			0.0209
28	30.0			0.0164
29	40.0			0.0143
30	50.0			0.0130
31	60.0	20.0	55.0	0.0120
32	70.0			0.0109
33	80.0			0.00876
34	90.0			0.00569
35	100.0			0.00287
36	110.0			0.00140

Table 1.2: Flux density in air (T)

$\Phi$ - $\Psi$  formulation

No	coordinates (mm)			B(T)
	x	y	z	
37	2.2	15.1	60.1	1.797
38	2.0	14.9	50.9	0.0287
39	1.5	0.0	55.0	0.517
40	1.5	0.0	25.0	1.349

Table 1.3: Flux densities in special points (T)  
 $\Phi$ - $\Psi$  formulation

No	Item	Specification
1	Code name	IGTEMAG3D
2	Formulation	FEM (Finite Element Method)
3	Governing equations	$\nabla \cdot (\mu \nabla \Phi) = \nabla \cdot (\mu \mathbf{H}_s)$ $\nabla \cdot (\mu \nabla \Psi) = 0$
4	Solution variables	$\Phi, \Psi$
5	Gauge condition	not applicable
6	Fraction of geometry	1/4
7	Technique for non-linear problem	Incremental method
	Convergence criterion	mean ( $\Delta \mu_r / \mu_r$ ) < 1% over all Gaussian points max ( $\Delta \mu_r / \mu_r$ ) < 5% over all Gaussian points
8	Approximation method of B-H curve	straight lines
9	Technique for open boundary problem	truncation
10	Calculation method of magnetic field produced by exciting current	Biot-Savart law (analytical) Biot-Savart law (numerical)
11	Property of coefficient matrix of linear equations	symmetric, sparse
12	Solution method for linear equations	ICCG
	Convergence criterion for iteration method	$\ Ax + b\ ^2 / \ b\ ^2 < 10^{-12}$
13	Element type	hexahedron nodal element (20 nodes)
14	Number of elements	48,384
15	Number of nodes	206,991
16	Number of unknowns	182,517
17	Computer	name: DECstation 5000-240 speed: 40 MIPS main memory: 264 MB precision of data: 64 bits CPU time total: 17,899 s

Table 1.4: Computational data,  $\Phi$ - $\Psi$  formulation

No.	coordinates (mm)			B(T)
	x	y	z	
1			0.0	1.419
2			10.0	1.406
3			20.0	1.377
4	0.0<x<1.6	-25.0<y<25.0	30.0	1.322
5			40.0	1.237
6			50.0	1.076
7			60.0	0.610
8	2.1			0.320
9	10.0			0.594
10	20.0			0.678
11	30.0			0.736
12	40.0			0.786
13	50.0	15.0<y<65.0	60.0<z<63.2	0.828
14	60.0			0.866
15	80.0			0.931
15	100.0			0.974
17	110.0			0.980
18	122.1			0.950
19			60.0	0.886
20			50.0	0.988
21			40.0	0.994
22	122.1<x<125.3	15.0<y<65.0	30.0	0.999
23			20.0	1.003
24			10.0	1.006
25			0.0	1.007

Table 2.1: Average flux densities in steel (T)  
T- $\Phi$  formulation

No	coordinates (mm)			B(T)
	x	y	z	
26	10.0			0.0348
27	20.0			0.0209
28	30.0			0.0163
29	40.0			0.0142
30	50.0			0.0130
31	60.0	20.0	55.0	0.0120
32	70.0			0.0108
33	80.0			0.00873
34	90.0			0.00568
35	100.0			0.00296
36	110.0			0.00141

Table 2.2: Flux density in air (T)  
T- $\Phi$  formulation

No	coordinates (mm)			B(T)
	x	y	z	
37	2.2	15.1	60.1	1.797
38	2.0	14.9	50.9	0.0287
39	1.5	0.0	55.0	0.517
40	1.5	0.0	25.0	1.349

Table 2.3: Flux densities in special points (T)  
T- $\Phi$  formulation

No	Item	Specification
1	Code name	IGTEMAG3D
2	Formulation	FEM (Finite Element Method)
3	Governing equations	$\nabla \cdot (\mu \nabla \Phi) = \nabla \cdot (\mu T)$ T represented by edge elements
4	Solution variables	$\Phi$
5	Gauge condition	not applicable
6	Fraction of geometry	1/4
7	Technique for non-linear problem	Incremental method
	Convergence criterion	mean ( $\Delta \mu_r / \mu_r$ ) < 1% over all Gaussian points max ( $\Delta \mu_r / \mu_r$ ) < 5% over all Gaussian points
8	Approximation method of B-H curve	straight lines
	Technique for open boundary problem	truncation
9		
10	Calculation method of magnetic field produced by exciting current	taking into account exciting current in governing equations directly
11	Property of coefficient matrix of linear equations	symmetric, sparse
12	Solution method for linear equations	ICCG
	Convergence criterion for iteration method	$\ Ax + b\ ^2 / \ b\ ^2 < 10^{-7}$
13	Element type	hexahedron nodal element (20 nodes) edge element (36 edges)
14	Number of elements	48,384
15	Number of nodes	206,991
16	Number of unknowns	182,517
17	Computer	name: DECstation 5000-240 speed: 40 MIPS main memory: 264 MB precision of data: 64 bits CPU time total: 13,907 s

Table 2.4: Computational data, T- $\Phi$  formulation

### A-edge formulation

This is a vector potential formulation without a gauge condition, using edge elements to represent  $\mathbf{A}$  [5]. The magnetic flux density is written as

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (7)$$

and the vector potential satisfies the differential equation

$$\nabla \times (\nu \nabla \times \mathbf{A}) = \mathbf{J} \quad (8)$$

The vector potential is approximated with the aid of edge elements and, in order to make the current density exactly divergence free, it is written in the form (6) with the same function  $T$  represented by edge elements used as in the  $T$ - $\Phi$  formulation.

The average flux density values in the three sections of the channel, along the specified line in the air and at the specified points are shown in Tables 3.1, 3.2 and 3.3. Some further information concerning the computation is summarized in Table 3.4.

No.	coordinates (mm)			B(T)
	x	y	z	
1	0.0 < x < 1.6	-25.0 < y < 25.0	0.0	1.344
2			10.0	1.333
3			20.0	1.299
4			30.0	1.241
5			40.0	1.152
6			50.0	1.015
7			60.0	0.677
8	2.1	15.0 < y < 65.0	60.0 < z < 63.2	0.270
9	10.0			0.556
10	20.0			0.640
11	30.0			0.700
12	40.0			0.749
13	50.0			0.792
14	60.0			0.830
15	80.0			0.895
16	100.0			0.939
17	110.0			0.945
18	122.1			0.950
19	122.1 < x < 125.3	15.0 < y < 65.0	60.0	0.951
20			50.0	0.954
21			40.0	0.959
22			30.0	0.964
23			20.0	0.968
24			10.0	0.971
25			0.0	0.972

Table 3.1: Average flux densities in steel (T)  
A-edge formulation

No	coordinates (mm)			B(T)
	x	y	z	
26	10.0	20.0	55.0	0.0344
27	20.0			0.0202
28	30.0			0.0162
29	40.0			0.0143
30	50.0			0.0130
31	60.0			0.0121
32	70.0			0.0108
33	80.0			0.00872
34	90.0			0.00573
35	100.0			0.00285
36	110.0			0.00144

Table 3.2: Flux density in air (T)  
A-edge formulation

No	coordinates (mm)			B(T)
	x	y	z	
37	2.2	15.1	60.1	1.524
38	2.0	14.9	50.9	0.0339
39	1.5	0.0	55.0	0.467
40	1.5	0.0	25.0	1.267

Table 3.3: Flux densities in special points (T)  
A-edge formulation

No	Item	Specification
1	Code name	IGTEMAG3D
2	Formulation	FEM (Finite Element Method)
3	Governing equations	$\nabla \times (\nu \nabla \times \mathbf{A}) = \mathbf{J}$
4	Solution variables	$\mathbf{A}$
5	Gauge condition	not imposed
6	Fraction of geometry	1/4
7	Technique for non-linear problem	Incremental method
	Convergence criterion	mean ( $\Delta \mu_r / \mu_r$ ) < 1% over all Gaussian points max ( $\Delta \mu_r / \mu_r$ ) < 5% over all Gaussian points
8	Approximation method of B-H curve	straight lines
9	Technique for open boundary problem	truncation
10	Calculation method of magnetic field produced by exciting current	taking into account exciting current in governing equations directly
11	Property of coefficient matrix of linear equations	symmetric, sparse
12	Solution method for linear equations	ICCG
	Convergence criterion for iteration method	$\ Ax + b\ ^2 / \ b\ ^2 < 10^{-7}$
13	Element type	hexahedron edge element (36 edges)
14	Number of elements	19,200
15	Number of nodes	84,083
16	Number of unknowns	225,728
17	Computer	name: DECstation 5000-240 speed: 40 MIPS main memory: 264 MB precision of data: 64 bits CPU time total: 50,412 s

Table 3.4: Computational data, A-edge formulation

## Results

The computed average flux densities in the steel channels and the flux density in the air are compared in Figs. 3 to 6 with the measured results [6]. Since the two scalar potential methods yield practically identical results, only a single curve is shown for this case in each plot. It seems that the values obtained by the vector potential formulation are somewhat nearer to the measurements in the steel but the deviation between the scalar potential and measured results is much less than it was reported in previous workshops for meshes with lower degrees of refinement [7-10].

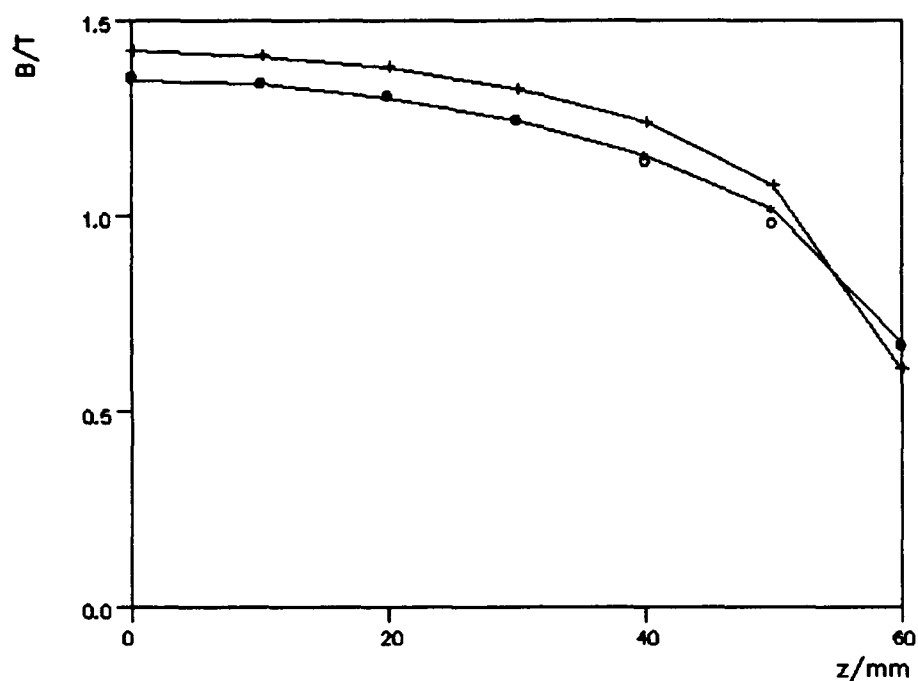


Fig. 3: Average flux density against  $z$ ,  $0 < x < 1.6$  mm,  $-25 < y < 25$  mm  
 o o o o: measurement, —+—+—: scalar potential, —\*—\*: vector potential

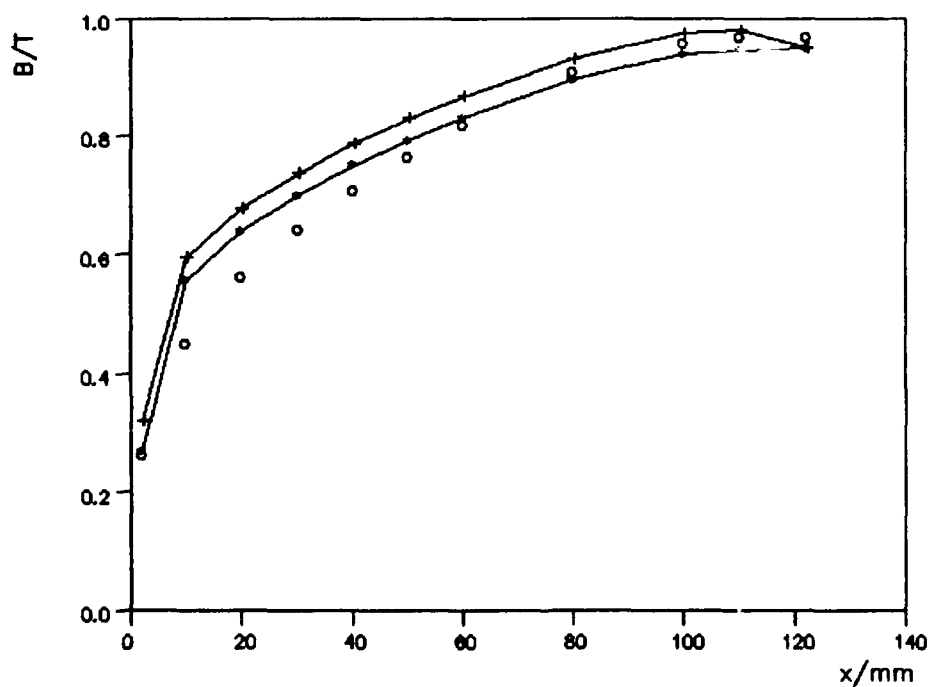


Fig. 4: Average flux density against  $x$ ,  $15 < y < 65$  mm,  $60 < z < 63.2$  mm  
 o o o o: measurement, —+—+—: scalar potential, —\*—\*: vector potential

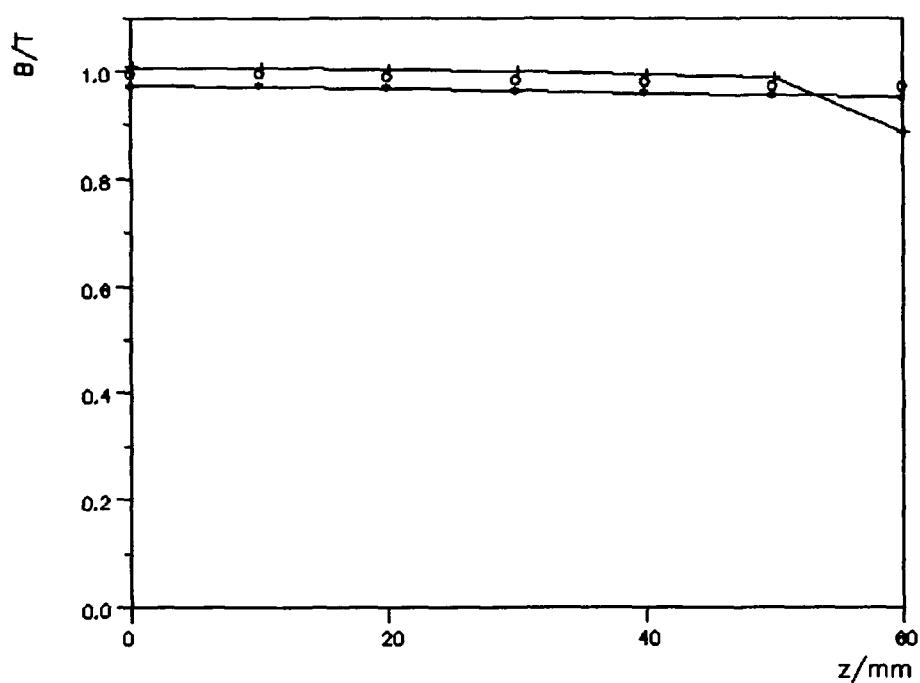


Fig. 5: Average flux density against  $z$ ,  $122.1 < x < 125.3$  mm,  $15 < y < 65$  mm  
 o o o o: measurement, —+—+—: scalar potential, —\*—\*—: vector potential

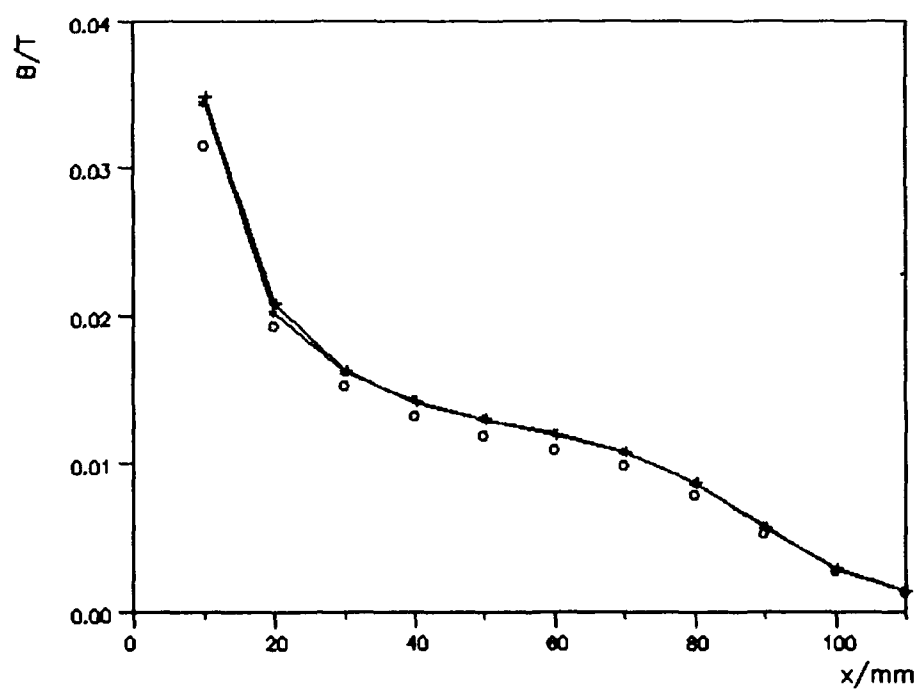


Fig. 6: Flux density against  $x$  along the line  $y = 20$  mm,  $z = 55$  mm  
 o o o o: measurement, —+—+—: scalar potential, —\*—\*—: vector potential

## References

- [1] T. Nakata and K. Fujiwara, "Summary of results for benchmark problem 13 (3-d nonlinear magnetostatic model)", in R. Albanese, E. Coccoresse, Y. Crutzen and P. Molino (ed.), *Proc., Third International TEAM Workshop*, Sorrento, Italy, 12-13 July 1991, pp. 223-249
- [2] J. Simkin and C. W. Trowbridge, "On the use of the total scalar potential in the numerical solution of field problems in electromagnetics", *Int. J. Numer. Meth. Eng.*, vol. 14, pp. 423-440, 1979.
- [3] C. J. Carpenter and E. A. Wyatt, "Efficiency of numerical techniques for computing eddy currents in two and three dimensions", *Proc., COMPUMAG*, Oxford, 31 March to 2 April 1976, pp. 242-250.
- [4] O. Biro, K. Preis, G. Vrsk, K.R. Richter and I. Tigar, "Computation of 3-D magnetostatic fields using a reduced scalar potential", *Fifth Biennial IEEE CEFC*, Claremont, CA, August 3-5, 1992. To be published in *IEEE Trans. on Magn.*, vol. 29, March 1993.
- [5] A. Kameari, "Calculation of transient 3D eddy current using edge-elements", *IEEE Trans. on Magn.*, vol. 26, pp. 466-469, 1990.
- [6] K. Fujiwara, private communication at the TEAM/ACES Workshop, Harvey Mudd College, 6-7 August, 1992.
- [7] Y. Crutzen, N.J. Diserens, C.R.I. Emson and D. Rodger (ed.), *Proc., European TEAM Workshop and International Seminar on Electromagnetic Field Analysis*, Oxford, England, 23-25 April 1990, pp. 107-163.
- [8] K.R. Richter, W.M. Rucker and O. Biro (ed.), *Proc., 4th International IGTE Symposium and European TEAM Workshop*, Graz, Austria, 10-12 October 1990, pp. 271-291.
- [9] L.R. Turner (ed.), *Proc., Toronto TEAM/ACES Workshop*, Ontario Hydro, 25 and 26 October 1990, ANL/FPP/TM-254, pp. 66-78.
- [10] R. Albanese, E. Coccoresse, Y. Crutzen and P. Molino (ed.), *Proc., Third International TEAM Workshop*, Sorrento, Italy, 12-13 July 1991, pp. 69-144 and pp. 223-249.



## **SOLUTION OF TEAM BENCHMARK PROBLEM #13 (3-D nonlinear magnetostatic model)**

Lauri Kettunen  
Tampere University of Technology  
P.O.Box 692  
33101 Tampere 10  
Finland

Larry R. Turner  
Argonne National Laboratory  
Advanced Photon Source  
Argonne, Illinois 60439  
U.S.A.

### **Abstract**

Four solutions for the TEAM magnetostatic benchmark #13 are presented. The problem was solved with the three dimensional volume integral code CORAL, formerly called GFUNET. A series of models were solved with increasing discretization in order to study the convergence and the charged CPU-time.

### **Problem Definition**

TEAM benchmark problem #13 is a magnetostatic problem consisting of a coil and steel plates. The geometry and all the material data is specified in details by Nakata, Takahashi, Fujiwara and Olszewski /1/. The purpose is to compute the magnetic flux density  $B$  along a given line outside the steel plates and in addition the average flux across certain planes inside the steel. Measured data are also provided. The geometry of the problem is shown in Fig. 1 and a schematic picture of the measurements in Fig. 2.

### **The Volume Integral Code CORAL**

The volume integral code CORAL is based on a decomposition of the magnetic field strength  $H$

$$H = H_m(M, r') + H_j(J, r').$$

$H_m$  is the field due to magnetization  $M$  and  $H_s$  the field due to currents. A system of integral equations is set up using a collocation approach and the line integrals of  $H$  along the edges of tetrahedra are solved /2/. The theoretical background of the formulation is explained in detail in reference /3/.

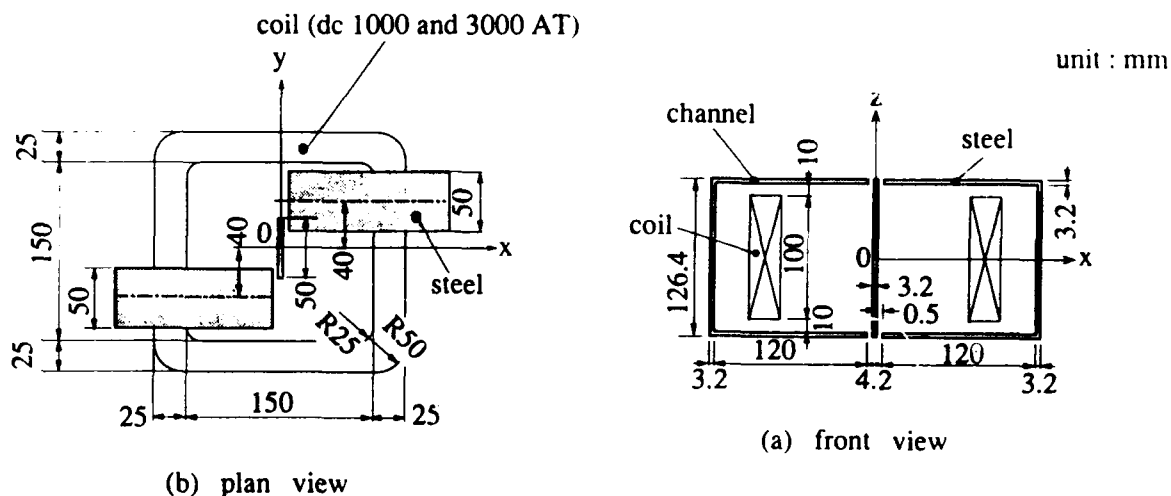


Figure 1. Geometry of the TEAM problem #13. /1/

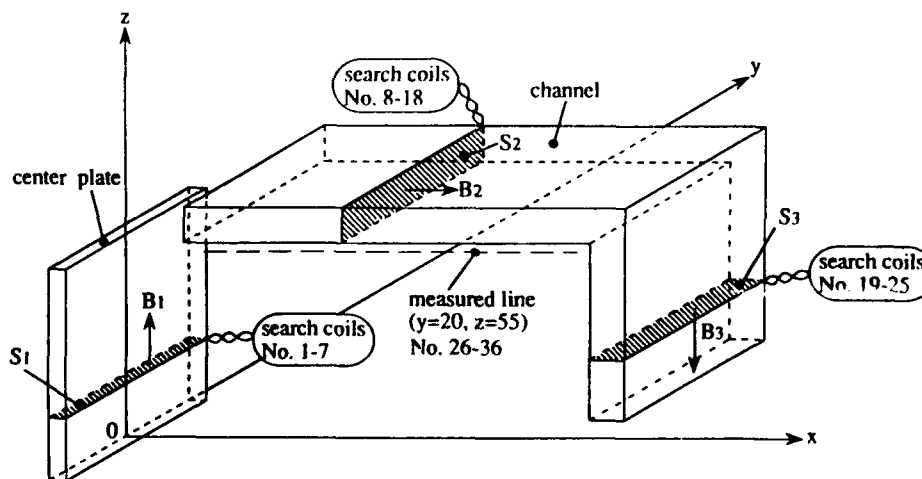


Figure 2. Schematic picture of the measurements. /1/

The main subroutines of CORAL are the integral equation matrix generation, the coil field computation routines, the solver, and the routines to update the susceptibility data and the matrix during a nonlinear iteration.

The first cycle of the nonlinear iteration, which sets up the matrix, takes always more CPU-time than the others, since vectors

$$C(r) = \int \frac{(r-r')}{\|r-r'\|^3} dv'$$

are generated only once and then stored on disc. These vectors are only geometry dependent and they are needed to compute the contribution of each tetrahedron to the scalar potential at each node. The so called paths /2//3/ are also created only once. They are, however, kept in the main memory all the time.

CORAL generates fairly large scratch files for the temporary data storage of the *C*-vectors. Vector *C* is integrated at each node from all the tetrahedra of the mesh. Thus the amount of disc space needed to store the *C* vectors is 3 x *nodes* x *tetrahedra* x *length of the variable* (i.e. 4 if single precision, 8 if double precision variables are used). At the moment, the size of the scratch file is the limiting factor we have reached preventing us of running very large problems. For instance using double precision variables a problem of 4950 tetrahedra and 2200 nodes generates a scratch file of 260 MB.

The integral equation matrix is diagonal dominant and nonsymmetric. Nothing else is known. Thus the solver we have used is based on LU-decomposition and backsubstitution. The lower and upper triangular are generated "in place" using Crout's algorithm with partial pivoting. The LU-decomposition requires about  $N^3/3$  operations and the backsubstitution stage  $N^2/2$  executions, where *N* is the number of equations. Hence the solution time of large problems increases rapidly and will cause problems in addition to the disc space needed.

At the moment the finite element mesh is generated by splitting hexahedra to five tetrahedra. This means that the interior tetrahedron of each hexahedron has the volume about twice as big as the others. It is not yet clear whether the bigger elements dominate the results or not. In the near future the present mesh generator will be changed to a 3D Delaunay tetrahedral mesh generator.

## Results

The results we presented in Sorrento workshop /4/ were solved with a fairly small workstation (24 MB SUN SPARCstation IPC), and hence it was dubious how the results changed if the mesh is refined. However, the integral formulation already seemed to share the same tendency as all other *h*-type formulations; the computed flux across the surfaces inside the steel plates is higher than the measured values.

The new results are obtained with a Kubota (Stardent) Titan 3010 computer. It has allowed tripling the number of tetrahedra and doubling the number of unknowns so far. The results seem to verify that with increasing discretization the convergence of the flux inside the steel plates is very slow; no obvious convergence was obtained. However, the magnetic field along the given line in air remained about the same all the time as shown in Fig. 2.

There has been debate on the reasons for the possible excessive values of flux in the steel plates. It is an interesting detail of the integral code that the only approximation made is the approximation of magnetic field  $H$  in the space  $W^1$  (the space spanned by the "edge elements"). In fact, the  $H$  is a vector field of  $W^1$ , which belongs to the class  $ker(curl)$ ; the closed line integrals of the field vanish. Thus, if the flux is too high, the problem seems to be related directly to the type of elements used.

The data of the four discretizations are shown in Table 1. The average flux in the steel plates is shown in Fig. 3. and the field in air in Fig. 4.

Table 1.

Case	Number of tetrahedra	Number of nodes	Number of equations
1	1505	720	718
2	3080	1104	1102
3	3705	1652	1650
4	3960	1380	1378

The charged CPU-time using the old version of CORAL varied between about 2000 to 42000 seconds. With the new version the solution time of the largest problem was about 31000 CPU-seconds. The elapsed times of the main subroutines of the new version are shown in Table 2.

Table 2.

Routine	Charged CPU-time	Percentage
Coil field computation	173.61 seconds	0.553 %
Generation of paths	6696.0 seconds	21.32 %
Matrix setup	1057.7 seconds	3.368 %
Solver	233.74 seconds/cycle 69 cycles	0.744 % / cycle 51.36 %

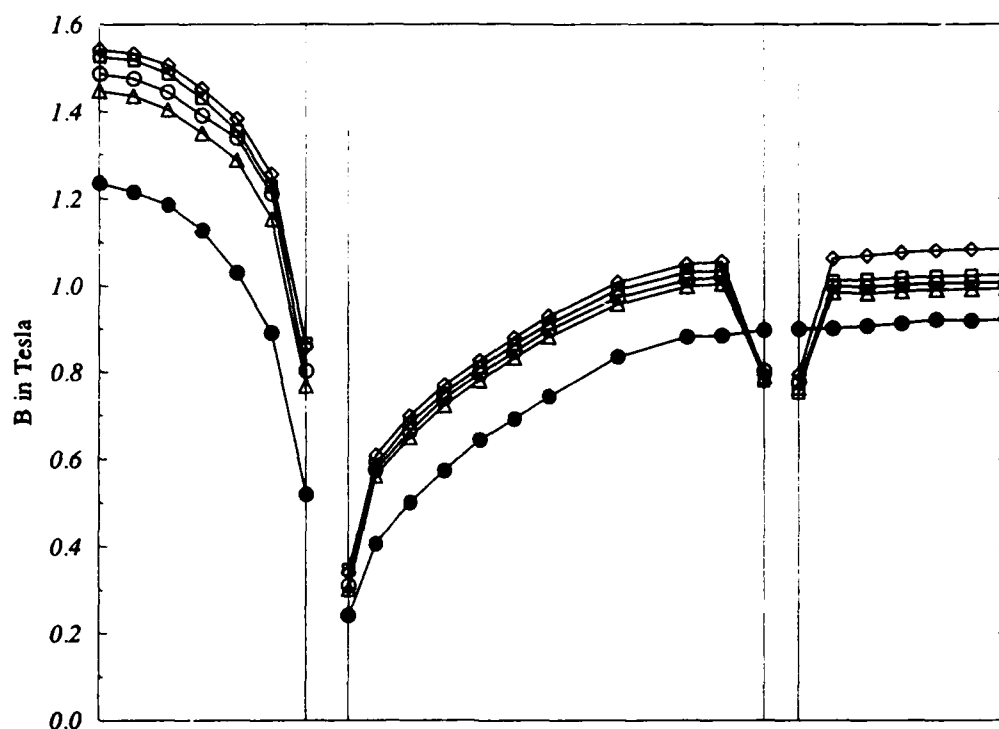


Figure 3. Average flux in the steel plates. Case 1, diamonds; case 2, squares; case 3, circles; case 4, triangles; measurements, filled circles.

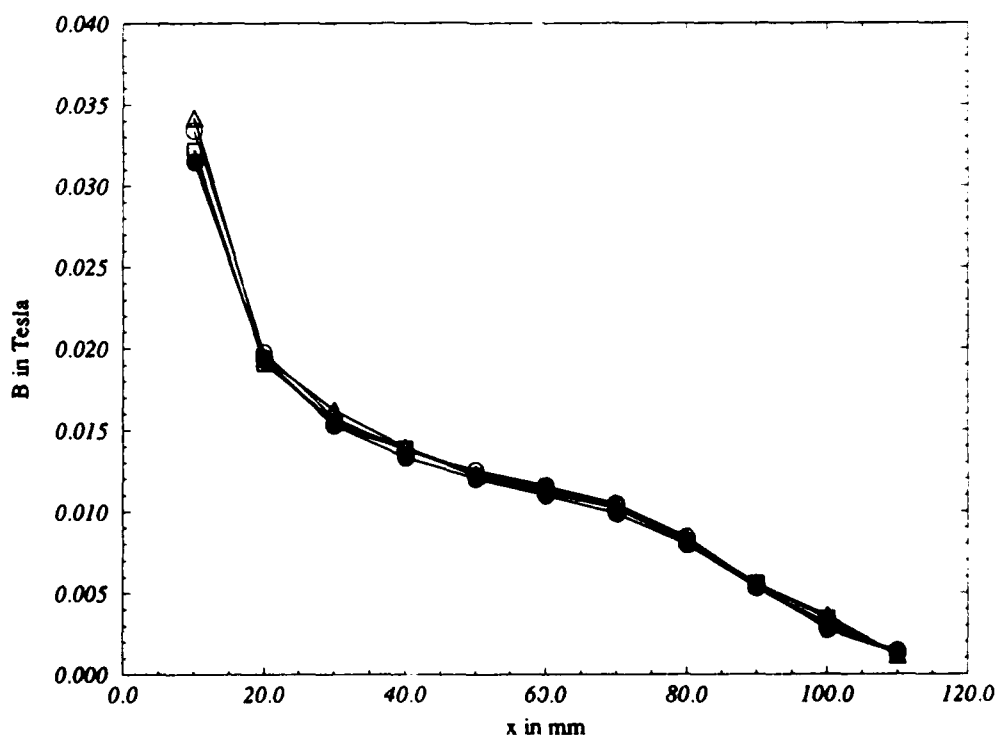


Figure 4. Magnetic field in air. Case 1, diamonds; case 2, squares; case 3, circles; case 4, triangles; measurements, filled circles.

## References

- /1/ T. Nakata, N. Takahashi, K. Fujiwara and P. Olszewski: Analysis of Magnetic Fields of 3-D Non-linear Magnetostatic Model (Problem 13), Proceedings of the European TEAM Workshop and International Seminar on Electromagnetic Field Analysis. Oxford, England, 23-25 April 1990. Commission of the European Communities, Joint Research Centre, EUR 12988 EN
- /2/ L. Kettunen and L.R. Turner: A Volume Integral Formulation for Nonlinear Magnetostatics and Eddy Currents Using Edge Elements, IEEE Trans, Magn., Vol. 28, March 1992, pp. 1639-1642
- /3/ Lauri Kettunen: Volume Integral Formulations for Three Dimensional Electromagnetic Field Computation, Tampere University of Technology, Publications 86, February 1992, Tampere University of Technology, Finland
- /4/ L. Kettunen and L.R. Turner: A Solution for TEAM Problem 13 Using an Integral Formulation for Nonlinear Magnetostatics, Proceedings of the Third International TEAM Workshop, Sorrento, Italy, 12-13 July 1991, Commission of the European Communities, Joint Research Centre, EUR 14173 EN

# SOLUTION OF TEAM BENCHMARK PROBLEM #9 Handling Velocity Effects with Variable Conductivity

Kent Davey  
School of Electrical Engineering  
Georgia Institute of Technology  
Atlanta, GA 30332-0250

## Abstract

Users often raise the question of whether it is possible to analyze eddy current problems with velocity effects within codes that are not programmed to account for movement. This paper looks at a technique for applying a conventional boundary element technique to the analysis of a velocity induced eddy current by altering the conductivity of the conducting medium as a function of position. Results of the predicted B fields for  $v=0$  m/s and  $v=10$  m/s are compared to the analytical solution of a coil traveling axially down the center of a conducting tube. Good agreement is achieved; further refinement could be realized by iterating on conductivity if necessary.

## The Boundary Element Approach

The problem to be analyzed is shown in Figure 1. The coil is excited at 50 Hz and is traveling down the pipe at velocity  $V$ . We analyze the problem with  $V=0$  m/s and 10 m/s. The boundary element approach (BEM) employed asks what fictitious free surface currents  $K_f$  could be placed on the skin of this pipe to account for the magnetization of the iron and the eddy currents. Actually 2 sets of surface currents are employed. A skin of currents just inside the pipe shell perimeter is used to represent the fields everywhere in the pipe. Another set of currents just outside the shell models the field in the air. The surface currents on the air side at  $r$  just less than 14 mm, dictate the field in the air region  $0 < r < 14$  mm. The surface currents just outside the skin at  $r=20$  mm, dictate the field for  $r > 20$  m. Once the surface currents are known, the magnetic field is found simply from Biot-Savart's law.

For the eddy current problem without movement, the pertinent equations for  $H$  and  $E$  are

$$\nabla \times \vec{H} = \sigma \vec{E} + \vec{J}_s \quad (1)$$

$$\vec{E} = -j\omega \vec{A} - \nabla \Phi \quad (2)$$

Writing (1) in terms of the vector potential  $A$  yields  
With the specified gauge of (3), the curl curl equation can be replaced by

$$\nabla \times \nabla \times \vec{A} - k^2 \vec{A} = \mu \vec{J}_s + \mu \sigma \nabla \Phi$$

where  $k^2 = j\omega\mu\sigma$ ,  
and  $\nabla \cdot \vec{A} = \mu\sigma\Phi$ .

(3)

$$\nabla^2 \vec{A} + k^2 \vec{A} = -\mu \vec{J}_s.$$
(4)

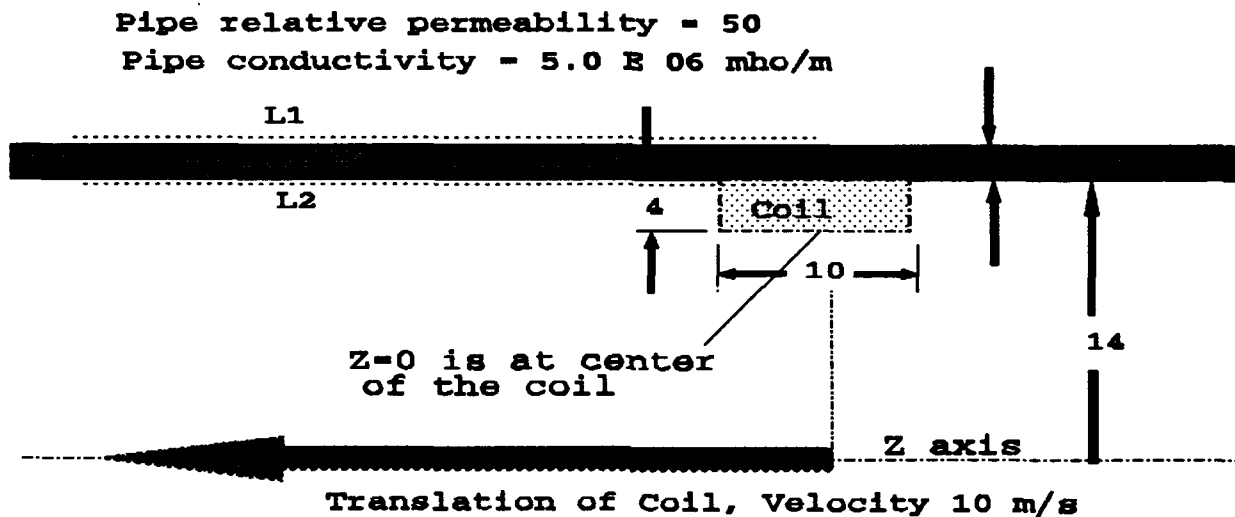


Figure 1 Coil traveling axially down a conducting pipe with velocity  $V$ . All dimensions are in millimeters. The coil is excited at 50 Hz. L1 and L2 are displaced 3 mm outside and inside the pipe respectively.

The integral solution for the vector potential due to a source current is <sup>1,2</sup>

$$A(r) = \mu \oint G(r, r') K_f(r') dS'.$$

where

$$G(r, r') = \frac{\mu}{2\pi} \int_0^\pi \frac{e^{jk|r-r'|}}{|r-r'|} dS'$$
(5)

Figure 2 helps to elucidate the approach. The fields in regions 1 and 2 are represented in terms of the surface currents and external impressed fields  $H_i$  and  $E_i$  as

$$\vec{H}^+ = \vec{H}_i + \vec{H}(K_f^+)$$
(6)

$$\vec{H}^- = \vec{H}(K_f^-)$$
(7)



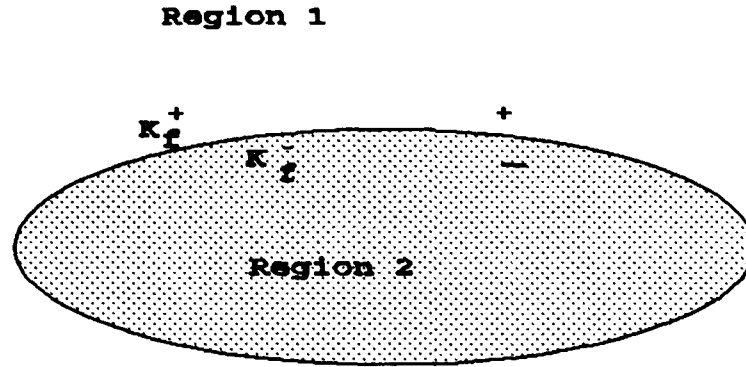


Figure 2 Two-region problem analyzed with BEM.

$$\vec{E}^+ = \vec{E}_1 + \vec{E}(K_f^+) \quad (8)$$

$$\vec{E}^- = \vec{E}(K_f^-) = -j\omega A^- \quad (9)$$

It only remains to impose the boundary conditions on E and H which are

$$\hat{n} \times (\vec{E}_2^+ - \vec{E}_1^-) = -\hat{n} \times \vec{E}_1 \quad (10)$$

$$\hat{n} \times (\vec{H}_2^+ - \vec{H}_1^-) = -\hat{n} \times \vec{H}_1 \quad (11)$$

Here  $\hat{n}$  is the outward normal to region 1. Note that the condition  $\hat{n} \cdot \vec{B} = 0$  is automatically insured by the use of the equivalent currents to directly compute B. Employing these boundary conditions yields the governing equations

$$j\omega \left[ \mu_2 \oint_{s'} G(k_2, r, r') K_f^+(r') dS' - \mu_1 \oint_{s'} G(k_1, r, r') K_f^-(r') dS' \right] = -E_t^1 = 0 \quad (12)$$

## Results

$$-\oint_{s'} K_f^+(r') \frac{\partial}{\partial n'} G(k2, r, r') dS' + \oint_{s'} K_f^-(r') \frac{\partial}{\partial n'} G(k1, r, r') dS' + 1/2 (K_f^+(r) + K_f^-(r)) = -\hat{n} \times \vec{H}_i \quad (13)$$

Equations (12) and (13) were applied to the problem both with the pipe having no relative permeability and with  $\mu_r=50$ . Both in this case and those to follow, 179 linear boundary elements were used, resulting in 366 unknowns. The field was predicted along the lines L1 and L2 of Figure 1. The radial and axial fields for the nonmagnetic pipe with the coil traveling at zero velocity are shown in Tables I and II.

Table I Radial Magnetic Fields  
nonmagnetic pipe, velocity = 0

z (mm)	Br on L1	Br on L1 analytic	Br on L2	Br on L2 analytic
0	1.05e-08		9.42e-08	
1.2	0.000184	0.000186	0.000824	0.000831
6	0.000688	0.000693	0.00345	0.00344
12	0.000627	0.000631	0.00119	0.0012
18	0.000396	0.000398	0.000532	0.000536
24	0.000237	0.000238	0.000264	0.000266
30	0.000143	0.000144	0.000142	0.000143
36	0.000089	0.00009	0.000081	0.000081
42	0.000057	0.000058	0.000049	0.000049
48	0.000038	0.000038	0.000031	0.000032
54	0.000026	0.000026	0.000021	0.000021
60	0.000018	0.000019	0.000014	0.000014
66	0.000013	0.000013	0.00001	0.00001
72	0.00001	0.00001	0.000007	0.000007

Table II Axial Magnetic Fields  
nonmagnetic pipe, velocity = 0

z	Bz on L1	Bz on L1 Analytic	Bz on L2	Bz on L2 Analytic
0	0.000889		0.00241	

1.2	0.000869	0.000876	0.00237	0.0022
6	0.000478	0.000481	0.000166	0.000109
12	0.00001	0.000004	0.000676	0.000681
18	0.000143	0.000144	0.000477	0.00048
24	0.000149	0.00015	0.000315	0.000318
30	0.000124	0.000125	0.000211	0.000212
36	0.000097	0.000098	0.000144	0.000145
42	0.000075	0.000075	0.000101	0.000102
48	0.000058	0.000058	0.000073	0.000074
54	0.000045	0.000045	0.000055	0.000055
60	0.000035	0.000036	0.000041	0.000042
66.00001	0.000028	0.000028	0.000032	0.000033
72	0.000023	0.000023	0.000025	0.000026

As expected, the ferromagnetic pipe with  $\mu_r=50$  has a diminished axial field on L1 outside the pipe. The radial and axial magnetic fields are shown compared to the analytic solution in Tables III and IV.

Table III Radial Magnetic Fields  
magnetic pipe, velocity = 0

z	Br on L1	Br on L1 analytic	Br on L2	Br on L2 analytic
0	7.94e-07		5.96e-07	
1.2	0.000015	0.000017	0.00131	0.00132
6	0.00006	0.000065	0.00539	0.0054
12	0.000062	0.000068	0.00154	0.00156
18	0.000048	0.000054	0.000515	0.000525
24	0.000037	0.000042	0.00018	0.000185
30	0.000029	0.000034	0.000064	0.000066
36	0.000024	0.000028	0.000023	0.000023
42	0.00002	0.000024	0.000008	0.000009
48	0.000017	0.000021	0.000003	0.000004
54	0.000015	0.000018	0.000001	0.000002

60	0.000013	0.000016	5.62e-07	6.65e-07
66.00001	0.000012	0.000015	3.61e-07	3.24e-07
72	0.00001	0.000013	2.87e-07	4.38e-07

Table IV Axial Magnetic Fields  
magnetic pipe, velocity = 0

z	Bz on L1	Bz on L1 Analytic	Bz on L2	Bz on L2 Analytic
0	0.000087		0.000459	
1.2	0.000085	0.000093	0.000449	0.00064
6	0.000055	0.000061	0.000306	0.000335
12	0.000016	0.000018	0.000079	0.000088
18	0.000004	0.000001	0.000029	0.000033
24	0.000006	0.000005	0.000012	0.000016
30	0.000007	0.000006	0.000007	0.00001
36	0.000007	0.000006	0.000005	0.000008
42	0.000006	0.000006	0.000004	0.000007
48	0.000006	0.000006	0.000004	0.000006
54	0.000005	0.000005	0.000003	0.000005
60	0.000005	0.000005	0.000003	0.000005
66.00001	0.000005	0.000005	0.000003	0.000005
72	0.000004	0.000004	0.000003	0.000004

### Velocity Effects

The remaining question is how to account for velocity effects. One alternative is to redefine the vector potential in terms of the axial velocity  $v$  of the pipe as <sup>3</sup>

$$\tilde{A} = Ae^{(-\frac{\mu_0 v z}{2})} \quad (14)$$

The governing equation becomes

$$\begin{aligned} \nabla^2 \tilde{A} - \alpha^2 \tilde{A} &= 0 \\ \text{where} \\ \alpha^2 &= \left( \frac{\mu \sigma v}{2} \right)^2 + j\omega \mu \sigma. \end{aligned} \quad (15)$$

Solution proceeds by solving for  $\tilde{A}$ .

The question in opening this paper seeks a solution without reformulating the program, i.e., using the same software as in the zero velocity case. We propose to trick the problem into thinking it is moving by altering the conductivity in front and to the rear of the coil. The defining vector potential equation with velocity is

$$\nabla^2 A - \mu \sigma \left( v \frac{\partial A}{\partial z} + \frac{\partial A}{\partial t} \right) = 0. \quad (16)$$

In cylindrical coordinates, this becomes

$$\frac{1}{\rho} \frac{\partial}{\partial \rho} \left( \rho \frac{\partial A}{\partial \rho} \right) + \frac{\partial^2 A}{\partial z^2} - \mu \sigma v \frac{\partial A}{\partial z} - \mu \sigma \frac{\partial A}{\partial t} - \frac{A}{\rho^2} = 0. \quad (17)$$

Terms 3 and 4 in (17) both share the common multiplier  $\sigma$ . One need merely to augment the conductivity to account for the effect of the velocity (term 3 in (17)). The steps for incorporating velocity are as follows:

1) Work the problem assuming  $v=0$ . Get  $A$  and  $\frac{\partial A}{\partial z}$  along the tube (wherever eddy currents exist)

2) Examine the ratio  $\left( \frac{v \frac{\partial A}{\partial z} + j\omega A}{j\omega A} \right)$ .

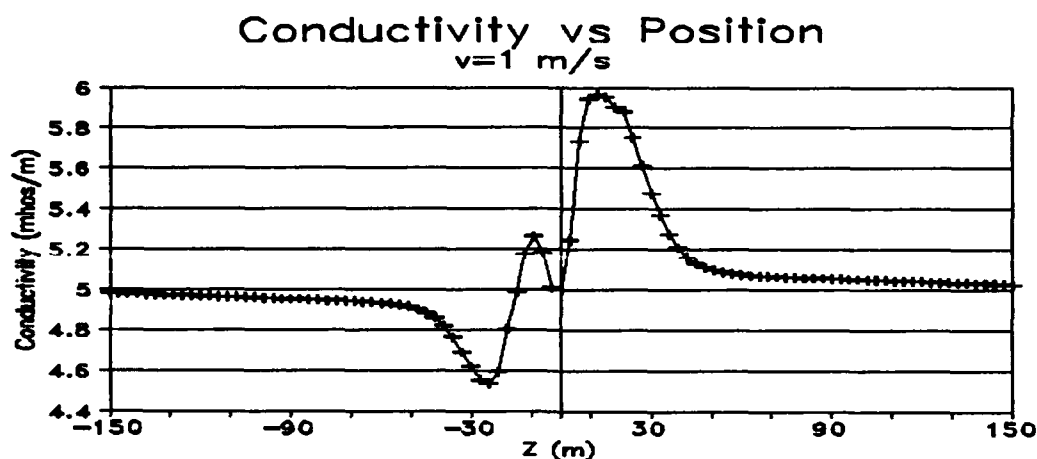
3) Increase the conductivity by the ratio

$$\sigma_{\text{new}} = \sigma_{\text{original}} \left| \frac{v \frac{\partial A}{\partial z} + j\omega A}{j\omega A} \right|.$$

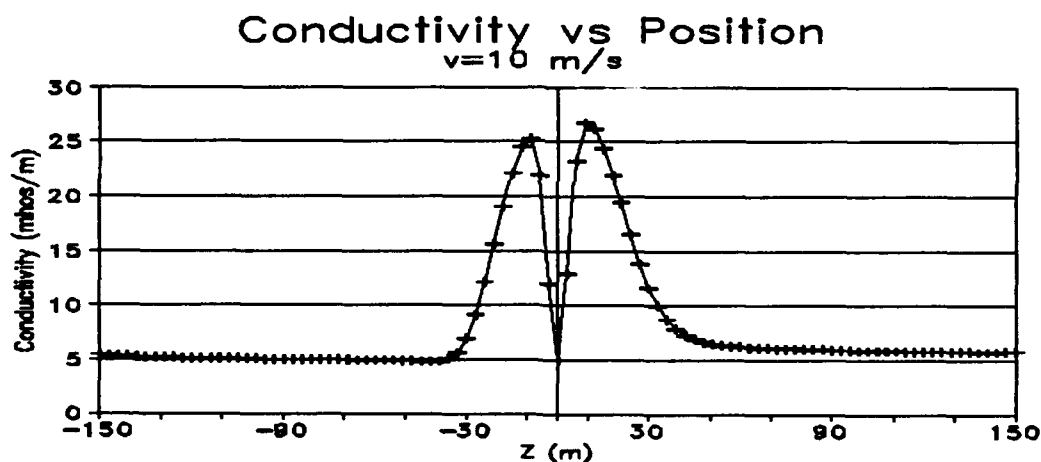
4). Repeat if necessary to refine the value of  $A$  and  $\frac{\partial A}{\partial z}$ .

Note that if the software used forces an entry of real conductivity as most do, the phase information of your final answer will not be correct. You are forced to use the absolute value of the ratio in step 3, but the magnitude should be correct.

Steps 1-3 were performed for problem 9 for the velocities  $v=1, 10$ , and  $100$  m/s. The conductivity profiles along the tube for these three velocities are shown in Figure 3, Figure 4, and Figure 5 respectively. Note that as the velocity is increased, the conductivity becomes more symmetric, indicating the overwhelming



*Figure 3 Conductivity in the tube for the  $v=1$  m/s velocity case.*



*Figure 4 Conductivity in the pipe for the  $v=10$  m/s velocity case.*

influence of the  $v \frac{\partial A}{\partial z}$  term compared to  $j\omega A$ .

#### Results for the $v=10$ m/s Velocity case

As seen in Figure 3, the effect of the velocity on the conductivity at  $v=1$  m/s is slight. The analytic results differed generally only in the second decimal place from the analytic results for the  $v=0$  m/s study. The  $v=10$  m/s case was on the other

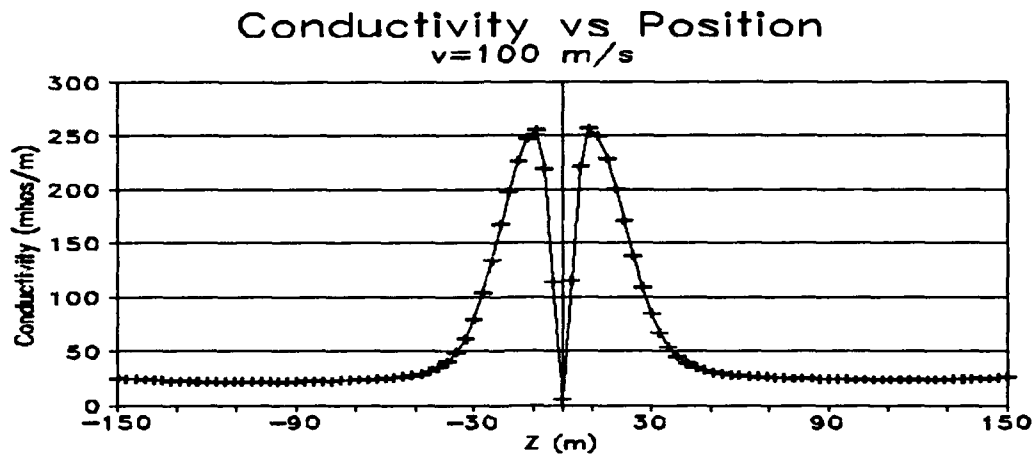


Figure 5 Conductivity in the tube for the  $v=100 \text{ m/s}$  velocity case.

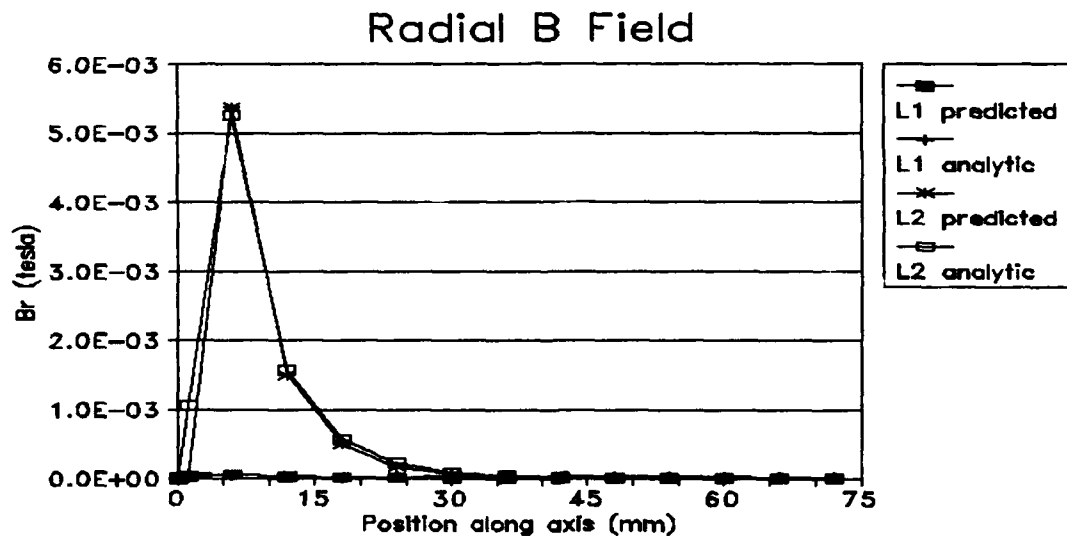


Figure 6 Radial field for the magnetic pipe,  $v=10 \text{ m/s}$ .

hand quite dissimilar. It was thought that this would prove a good testing ground for the theory. Shown in Figure 6 is the radial field predicted along L1 and L2 with its analytic counterpart. The tabular comparison is shown in Table V.

Table V Radial Field Predictions  
 $v=10$  m/s, permeability = 50

z	Br on L1	Br on L1 analytic	Br on L2	Br on L2 analytic
0	0.000004	0	0.000007	0
1.2	0.000027	0.000059	0.000013	0.00106
6	0.000057	0.000055	0.005381	0.00525
12	0.000032	0.000028	0.001499	0.00157
18	0.00002	0.000005	0.000492	0.000561
24	0.000014	0.000023	0.00017	0.000217
30	0.000011	0.000031	0.000059	0.000089
36	0.000009	0.000032	0.000021	0.000039
42	0.000007	0.000031	0.000007	0.000019
48	0.000006	0.000028	0.000003	0.00001
54	0.000006	0.000026	8.27e-07	0.000006
60	0.000005	0.000023	2.42e-07	0.000003
66.00001	0.000004	0.000021	6.19e-08	0.000002
72	0.000004	0.000019	5.93e-08	0.000002

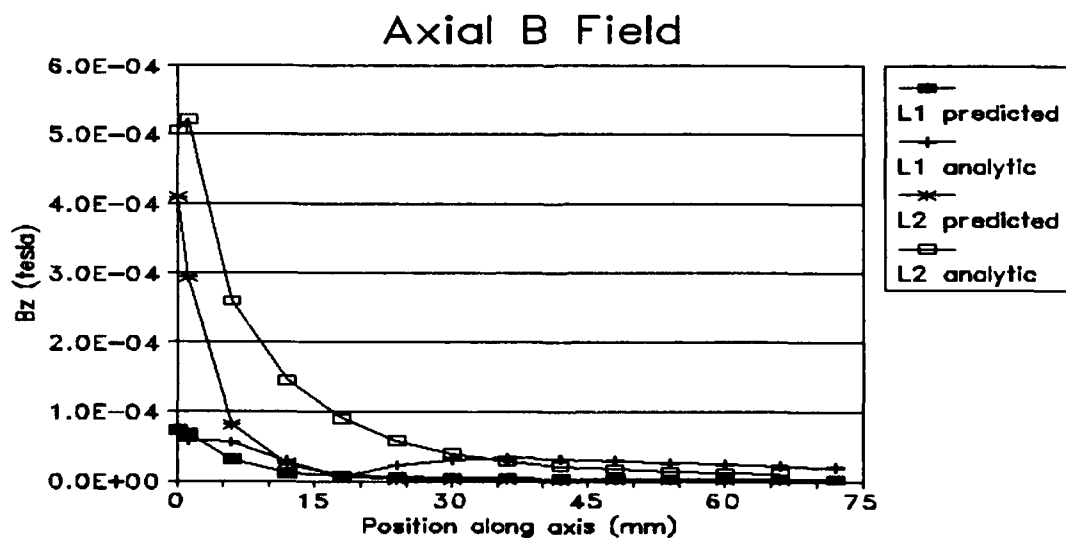


Figure 7 Axial magnetic field for the magnetizable pipe with coil traveling at  $v=10$  m/s.



By comparison Figure 7 shows the axial B field predictions along with those obtained analytically. In both cases some error is seen in the smallest component of the field, but the resultant is very close. Table Vi displays this data along with the analytic predictions.

Table VI Axial Field Predictions  
v=10 m/s, permeability = 50

z	Bz on L1	Bz on L1 analytic	Bz on L2	Bz on L2 analytic
0	0.000072		0.00042	
1.2	0.000068	0.000059	0.000409	0.000506
6	0.00003	0.000055	0.000293	0.000522
12	0.000011	0.000028	0.00008	0.000259
18	0.000007	0.000005	0.000025	0.000146
24	0.000006	0.000023	0.000007	0.000089
30	0.000004	0.000031	0.000002	0.000057
36	0.000004	0.000032	0.000001	0.000039
42	0.000003	0.000031	0.000001	0.000028
48	0.000002	0.000028	0.000001	0.000021
54	0.000002	0.000026	0.000001	0.000016
60	0.000002	0.000023	0.000001	0.000013
66.00001	0.000002	0.000021	0.000001	0.000011
72	0.000002	0.000019	9.20e-07	0.000009

The accuracy suggests that the method is quite effective.

### Conclusions

Altering the conductivity to account for velocity effects is a relatively simple technique for accounting for velocity when the code does not implicitly have such capability. In this example, the conductivity was altered in the tube in regions to be piecewise continuous. Only 14 different conductivities were used to model

Figure 4. Furthermore the ratio  $\sigma_{new} = \sigma_{original} \left| \frac{v \frac{\partial A}{\partial z} + j\omega A}{j\omega A} \right|$  was computed in the center of the pipe at the radial line r=17 mm. In reality 3 further modifications would be necessary to get precise results.

1) Alter the conductivity to reflect radial changes in the ratio

$$\sigma_{new} = \sigma_{original} \left| \frac{v \frac{\partial A}{\partial z} + j\omega A}{j\omega A} \right| \dots$$

2). Model a continuous change in conductivity as suggested by Figure 4.

3). Iterate on the solution to refine the conductivities with a

closer estimate of  $\sigma_{new} = \sigma_{original} \left| \frac{v \frac{\partial A}{\partial z} + j\omega A}{j\omega A} \right|$  after the first iteration.

The accuracy of the answers reflects the fact that the ratio does not change significantly as one varies the velocity. Also reasonable predictions of the fields are realized with a rather crude modeling of the conductivity.

If a complex conductivity is known, it can be inserted to correctly account for the  $v \frac{\partial A}{\partial z}$  term. Since this is unknown a priori, one is forced to iteratively approach its correct value. The problem is worked first assuming it is zero, and then updating the value as suggested above. The accuracy of the results summarized below were obtained in a single iteration. They enable the user to obtain a close result without reformulating the Green's function integral. Many users do not have access to the code to make these alterations even if they could formulate the changes.

### Acknowledgements

The BEM package used for these calculations was a program called Oersted from Integrated Engineering Software. The author wishes to thank Dalian Zheng for his help in checking the  $v=0$  computations.

### References

1. Smythe, Static and Dynamic Electricity, McGraw Hill, 1965, p 290.
2. M. Burnet-Fauchez and R. Michaux, "Boundary Element Calculation of 2D magnetic fields with eddy currents", Modelec Conf. La Grande Motte, Oct. 1984, Pluralis Ed. Paris, pp 97-109.
3. G. Demey, "Field Calculations in moving conductors by an integral equation", Archiv fur elektrotechnik, vol 64, 1981, pp 101-103.

## 1993 INSTITUTIONAL MEMBERS

THE AEROSPACE CORPORATION PO Box 92957 Los Angeles, CA 90245-2957	ANDREW CORPORATION 10500 W 153rd Street Orland Park, IL 60462	ASELSAN INCORPORATED PO Box 101 Yenimahalle Ankara, TURKEY 06172
ATOMIC WEAPONS ESTAB UK Building E3, Ave Aldermaston Reading, Berkshire, UK RG7 4PR	B.M.S. COMPUTER SYSTEMS Sproughton House Ipswich, Suffolk, UK IP8 3AW	BOEING COMPANY PO Box 3707 Seattle, WA 98124-2207
BOLT, BERANEK & NEWMAN INC. 10 Moulton Street Cambridge, MA 02138	COMPAQ COMPUTER CORPORATION PO Box 692000 Houston, TX 77269-2000	CSELT-RADIO Via Guglielmo Reiss Romoli 274 Turin, ITALY 10148
CULHAM LAB UK Atomic Energy Authority Abingdon, Oxfordshire, UK OX14 3DB	D L R OBERPFAFFENHOFEN Zentralbibliothek Wessling, GERMANY 8031	D S T O SALISBURY LIBRARY Box 1500 Salisbury, SA, AUSTRALIA 5108
DAWSON'S SUBSCRIPTION SERVICE PO Box 191 Mt Morris, IL 61054-0191	DEFENCE TECH & PROCUREMENT NC- Lab Spiez, SWITZERLAND 3700	DEFENSE RESEARCH ESTAB. 1100 Carling Avenue Ottawa, On, CANADA K1A 0K2
DSTO Private Bag 12 Ascot Vale, Victoria, AUSTRALIA 3032	DYNETICS, INCORPORATED PO Drawer B Huntsville, AL 35814-5050	ECOLE SUPERIEURE D'ELECT. Plateau du Moulon 91192 Gif-Sur-Yvette, Cedex, FRANCE
ELECTRONICS RESEARCH INC. 108 Market Street Newburgh, IN 47630	ERIM INFORMATION PO Box 134001 Ann Arbor, MI 48113	FGAN/FHP Neuenahrer Strabe 20 D-5307 Wachtberg-Werthoven, GERMANY
FOKKER AIRCRAFT B.V. Postbus 7600/Infotheek Schiphol, NETHERLANDS 1117 ZJ	FORSCHUNGS FUR INFOTECH MBH PO Box 1147 Bad Salzdetfurth, GERMANY W 3202	GRUMMAN CORP. RES. CENTER Dept. 0582 & 0589 Bethpage, NY 11714-3580
HARRIS CORPORATION 1680 University Avenue Rochester, NY 14610-9983	HATFIELD & DAWSON 4226 Sixth Avenue, NW Seattle, WA 98107	HOKKAIDO DAIGAKU Nishi 8, Kita 13 Sapporo, JAPAN 060
HONG KONG UNIVERSITY Tai PO Tsai, Clear Water Bay Road Kowloon, HONG KONG	HUGHES RESEARCH LIBRARY 3011 Malibu Canyon Road Malibu, CA 90255-4797	HUNTING ENGINEERING LTD. Redding Wood Amphill, Bedford, UK MK45 2HD
IIT RESEARCH INSTITUTE 185 Admiral Cochrane Dr. Annapolis, MD 21401-7347	IMAGINEERING LIMITED 95 Barber Greene Rd, Suite 112 Don Mills, Ontario, CANADA M3C 3E9	IMPERIAL COLLEGE Exhibition Road London, UK SW7 2BX
KATHREIN-WERKE KG Postbox 260 Rosenheim 2, GERMANY D-8200	KERSHNER, WRIGHT & HAGAMAN 5730 General Washington Drive Alexandria, VA 22312	LINDA HALL LIBRARY 5109 Cherry Street Kansas City, MO 64110
LITTON ELECTRON DEVICES 960 Industrial Road San Carlos, CA 94404	LOCKHEED MISSILES AND SPACE 3251 Hanover Street Palo Alto, CA 94304	MARCONI DEFENCE SYSTEMS The Grove, Warren Lane Stanmore, Middlesex, UK HA7 4LY
MIT LINCOLN LABORATORY LIB. 244 Wood Street Lexington, MA 02173-0073	MONASH UNIVERSITY/CAULFIELD PO Box 197/Caulfield East Melbourne, Victoria, AUSTRALIA 3145	MOTOROLA INCORPORATED 1301 E Algonquin Road Schamburg, IL 60196
MPB TECHNOLOGIES 151 Hymus Boulevard H9R 1E9 Pointe-Claire, Quebec, CANADA	NORWEGIAN TELECOM RESEARCH PO Box 83 Kjeller, NO. PWAY N-2007	NOVATEL COMMUNICATIONS 1020 64th Avenue NE Calgary, Alberta, CANADA T2E 7V8
OCEAN APPLIED RES. CORP. 10447 Roselle Street San Diego, CA 92121	OFFICE OF RESEARCH PO Box 15183 Arlington, VA 22215-0183	OHIO UNIVERSITY 325 Stocker Center Athens, OH 45701

### 1993 INSTITUTIONAL MEMBERS (Continued)

PARAMAX SYSTEMS CANADA 6111 Royalmount Avenue Montreal, Que. CANADA H4P1K6	PENNSYLVANIA STATE UNIVERSITY Pattee Library University Park, PA 16802	PETERS, ARTHUR K. CONSULTING 7020 N.W. 11th Place Gainesville, FL 32605
POTCHEFSTROOM UNIVERSITY PU for CHE/EE Department Potchefstroom, S AFRICA 2520	QUEEN MARY & WESTFELD COLL Mile End Road London, UK E1 4NS	RADIO FREQUENCY SYSTEMS Box 191 3136 Croydon, Victoria, AUSTRALIA
ROCKWELL INTERNATIONAL 1745 Jeff Davis Highway Arlington, VA 22202	SAAB--SCANIA MILITARY AIRCRAFT Tusta 5 Linkoping, SWEDEN S-58188	SAIC 300 Nickerson Road Marlborough, MA 01752
STG NATIONAL LUCHT-EM Ruimtevaart Lab. Anthony Fokkerweg 2 Amsterdam, NETHERLANDS 1059	SWETS SUBSCRIPTION SERVICE 650 Swedesford Road PO Box 517 Berwyn, PA 19312	SWISS FEDERAL INST OF TECH ETH Zentrum, ETZ G94 Zurich, SWITZERLAND CH-8092
SYRACUSE RESEARCH CORPORATION Merrill Lane Syracuse, NY 13210-4080	TASC 55 Walkers Brook Drive Reading, MA 01887	TECHNION-ISRAEL INST OF TECH Faculty of Electrical Engineering Technion City, ISRAEL 32000
TECHNICAL RESEARCH CENTRE Telecom Lab/Otakaari 7B Espoo, FINLAND SF-01250	TELECOM AUSTRALIA 770 Blackburn Road Clayton, Victoria, AUSTRALIA 3168	TELEX COMMUNICATIONS INC. 8601 E. Cornhusker Highway Lincoln, NE 68505
UNIVERSITY OF CENTRAL FLORIDA PO Box 162440 Orlando, FL 32816	UNIVERSITAT DER BUNDESWEHR Werner Heisenberg Weg 39 Neubiberg, GERMANY W-8014	VISTA RESEARCH INC. 100 View St, #202, Box 998 Mountain View, CA 94042

## ACES COPYRIGHT FORM

This form is intended for original, previously unpublished manuscripts submitted to ACES periodicals and conference publications. The signed form, appropriately completed, MUST ACCOMPANY any paper in order to be published by ACES. PLEASE READ REVERSE SIDE OF THIS FORM FOR FURTHER DETAILS.

TITLE OF PAPER:

AUTHORS(S)

PUBLICATION TITLE/DATE:

RETURN FORM TO:

Dr. Richard W. Adler  
Naval Postgraduate School  
Code EC/AB  
Monterey, CA 93943

---

### PART A - COPYRIGHT TRANSFER FORM

(NOTE: Company or other forms may not be substituted for this form. U.S. Government employees whose work is not subject to copyright may so certify by signing Part B below. Authors whose work is subject to Crown Copyright may sign Part C overleaf).

The undersigned, desiring to publish the above paper in a publication of ACES, hereby transfer their copyrights in the above paper to The Applied Computational Electromagnetics Society (ACES). The undersigned hereby represents and warrants that the paper is original and that he/she is the author of the paper or otherwise has the power and authority to make and execute this assignment.

Returned Rights: In return for these rights, ACES hereby grants to the above authors, and the employers for whom the work was performed, royalty-free permission to:

1. Retain all proprietary rights other than copyright, such as patent rights.
2. Reuse all or portions of the above paper in other works.
3. Reproduce, or have reproduced, the above paper for the author's personal use or for internal company use provided that (a) the source and ACES copyright are indicated, (b) the copies are not used in a way that implies ACES endorsement of a product or service of an employer, and (c) the copies per se are not offered for sale.
4. Make limited distribution of all or portions of the above paper prior to publication.
5. In the case of work performed under U.S. Government contract, ACES grants the U.S. Government royalty-free permission to reproduce all or portions of the above paper, and to authorize others to do so, for U.S. Government purposes only.

ACES Obligations: In exercising its rights under copyright, ACES will make all reasonable efforts to act in the interests of the authors and employers as well as in its own interest. In particular, ACES REQUIRES that:

1. The consent of the first-named author be sought as a condition in granting re-publication permission to others.
2. The consent of the undersigned employer be obtained as a condition in granting permission to others to reuse all or portions of the paper for promotion or marketing purposes.

In the event the above paper is not accepted and published by ACES or is withdrawn by the author(s) before acceptance by ACES, this agreement becomes null and void.

---

AUTHORIZED SIGNATURE

TITLE (IF NOT AUTHOR)

---

EMPLOYER FOR WHOM WORK WAS PERFORMED

DATE FORM SIGNED

### PART B - U.S. GOVERNMENT EMPLOYEE CERTIFICATION

(NOTE: If your work was performed under Government contract but you are not a Government employee, sign transfer form above and see item 5 under Returned Rights).

This certifies that all authors of the above paper are employees of the U.S. Government and performed this work as part of their employment and that the paper is therefore not subject to U.S. copyright protection.

---

AUTHORIZED SIGNATURE

TITLE (IF NOT AUTHOR)

---

NAME OF GOVERNMENT ORGANIZATION

DATE FORM SIGNED

---

## PART C - CROWN COPYRIGHT

(Note: ACES recognizes and will honor Crown Copyright as it does U.S. Copyright. It is understood that, in asserting Crown Copyright, ACES in no way diminishes its rights as publisher. Sign only if ALL authors are subject to Crown Copyright.

This certifies that all authors of the above Paper are subject to Crown Copyright. (Appropriate documentation and instructions regarding form of Crown Copyright notice may be attached).

---

AUTHORIZED SIGNATURE

TITLE OF SIGNED

---

NAME OF GOVERNMENT BRANCH

DATE FORM SIGNED

### Information to Authors

#### ACES POLICY

ACES distributes its technical publications throughout the world, and it may be necessary to translate and abstract its publications, and articles contained therein, for inclusion in various compendiums and similar publications, etc. When an article is submitted for publication by ACES, acceptance of the article implies that ACES has the rights to do all of the things it normally does with such an article.

In connection with its publishing activities, it is the policy of ACES to own the copyrights in its technical publications, and to the contributions contained therein, in order to protect the interests of ACES, its authors and their employers, and at the same time to facilitate the appropriate re-use of this material by others.

The new United States copyright law requires that the transfer of copyrights in each contribution from the author to ACES be confirmed in writing. It is therefore necessary that you execute either Part A-Copyright Transfer Form or Part B-U.S. Government Employee Certification or Part C-Crown Copyright on this sheet and return it to the Managing Editor (or person who supplied this sheet) as promptly as possible.

#### CLEARANCE OF PAPERS

ACES must of necessity assume that materials presented at its meetings or submitted to its publications is properly available for general dissemination to the audiences these activities are organized to serve. It is the responsibility of the authors, not ACES, to determine whether disclosure of their material requires the prior consent of other parties and if so, to obtain it. Furthermore, ACES must assume that, if an author uses within his/her article previously published and/or copyrighted material that permission has been obtained for such use and that any required credit lines, copyright notices, etc. are duly noted.

#### AUTHOR/COMPANY RIGHTS

If you are employed and you prepared your paper as a part of your job, the rights to your paper initially rest with your employer. In that case, when you sign the copyright form, we assume you are authorized to do so by your employer and that your employer has consented to all of the terms and conditions of this form. If not, it should be signed by someone so authorized.

NOTE RE RETURNED RIGHTS: Just as ACES now requires a signed copyright transfer form in order to do "business as usual", it is the intent of this form to return rights to the author and employer so that they too may do "business as usual". If further clarification is required, please contact: The Managing Editor, R.W. Adler, Naval Postgraduate School, Code EC/AB, Monterey, CA, 93943, USA (408)646-2352.

Please note that, although authors are permitted to re-use all or portions of their ACES copyrighted material in other works, this does not include granting third party requests for reprinting, republishing, or other types of re-use.

#### JOINT AUTHORSHIP

For jointly authored papers, only one signature is required, but we assume all authors have been advised and have consented to the terms of this form.

#### U.S. GOVERNMENT EMPLOYEES

Authors who are U.S. Government employees are not required to sign the Copyright Transfer Form (Part A), but any co-authors outside the Government are.

Part B of the form is to be used instead of Part A only if all authors are U.S. Government employees and prepared the paper as part of their job.

NOTE RE GOVERNMENT CONTRACT WORK: Authors whose work was performed under a U.S. Government contract but who are not Government employees are required to sign Part A-Copyright Transfer Form. However, item 5 of the form returns reproduction rights to the U.S. Government when required, even though ACES copyright policy is in effect with respect to the reuse of material by the general public.

## INFORMATION FOR AUTHORS

### PUBLICATION CRITERIA

Each paper is required to manifest some relation to applied computational electromagnetics. Papers may address general issues in applied computational electromagnetics, or they may focus on specific applications, techniques, codes, or computational issues. While the following list is not exhaustive, each paper will generally relate to at least one of these areas:

1. Code validation. This is done using internal checks or experimental, analytical or other computational data. Measured data of potential utility to code validation efforts will also be considered for publication.

2. Code performance analysis. This usually involves identification of numerical accuracy or other limitations, solution convergence, numerical and physical modeling error, and parameter tradeoffs. However, it is also permissible to address issues such as ease-of-use, set-up time, run time, special outputs, or other special features.

3. Computational studies of basic physics. This involves using a code, algorithm, or computational technique to simulate reality in such a way that better or new physical insight or understanding is achieved.

4. New computational techniques, or new applications for existing computational techniques or codes.

5. "Tricks of the trade" in selecting and applying codes and techniques.

6. New codes, algorithms, code enhancement, and code fixes. This category is self-explanatory but includes significant changes to existing codes, such as applicability extensions, algorithm optimization, problem correction, limitation removal, or other performance improvement. NOTE: CODE (OR ALGORITHM) CAPABILITY DESCRIPTIONS ARE NOT ACCEPTABLE, UNLESS THEY CONTAIN SUFFICIENT TECHNICAL MATERIAL TO JUSTIFY CONSIDERATION.

7. Code input/output issues. This normally involves innovations in input (such as input geometry standardization, automatic mesh generation, or computer-aided design) or in output (whether it be tabular, graphical, statistical, Fourier-transformed, or otherwise signal-processed). Material dealing with input/output data base management, output interpretation, or other input/output issues will also be considered for publication.

8. Computer hardware issues. This is the category for analysis of hardware capabilities and limitations in meeting various types of electromagnetics computational requirements. Vector and parallel computational techniques and implementation are of particular interest.

Applications of interest include, but are not limited to, antennas (and their electromagnetic environments), networks, static fields, radar cross section, shielding, radiation hazards, biological effects, electromagnetic pulse (EMP), electromagnetic interference (EMI), electromagnetic compatibility, power transmission, charge transport, dielectric and magnetic materials, microwave components, MMIC technology, remote sensing and geophysics, communications systems, fiber optics, plasmas, particle accelerators, generators and motors, electromagnetic wave propagation, non-destructive evaluation, eddy currents, and inverse scattering.

Techniques of interest include frequency-domain and time-domain techniques, integral equation and differential equation techniques, diffraction theories, physical optics, moment methods, finite differences and finite element techniques, modal expansions, perturbation methods, and hybrid methods. This list is not exhaustive.

A unique feature of the Journal is the publication of unsuccessful efforts in applied computational electromagnetics. Publication of such material provides a means to discuss problem areas in electromagnetic modeling. Material representing an unsuccessful application or negative results in computational electromagnetics will be considered for publication only if a reasonable expectation of success (and a reasonable effort) are reflected. Moreover, such material must represent a problem area of potential interest to the ACES membership.

### EDITORIAL REVIEW

In order to ensure an appropriate level of quality control, papers are refereed. They are reviewed both for technical correctness and for adherence to the listed guidelines regarding information content. Authors should submit the initial manuscript in draft form so that any suggested changes can be made before the photo-ready copy is prepared.